

Los más grandes logros de la humanidad, no siempre se alcanzan con el consenso de todos.

Cristóbal Colón.

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE CONTADURÍA PÚBLICA Y ADMINISTRACIÓN

DIVISIÓN DE ESTUDIOS DE POSTGRADO



APLICACIÓN DE UN ALGORITMO GENÉTICO GENERACIONAL PARA UN PROBLEMA DE LOGÍSTICA DE RUTEO CON ENTREGA Y RECOLECCION DE PRODUCTO Y CON VENTANAS DE HORARIO NEGOCIABLES.

Presentada por:

JESÚS FABIÁN LÓPEZ PÉREZ

Disertación enviada como requisito parcial para obtener el Grado de DOCTOR EN FILOSOFÍA CON ESPECIALIDAD EN ADMINISTRACIÓN.

San Nicolás de los Garza, Nuevo León, MAYO 2004.

© Copyright 2004. Todos los derechos reservados.

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE CONTADURÍA PÚBLICA Y ADMINISTRACIÓN

DIVISION DE ESTUDIOS DE POSTGRADO

**APLICACIÓN DE UN ALGORITMO GENÉTICO GENERACIONAL PARA UN PROBLEMA
DE LOGÍSTICA DE RUTEO CON ENTREGA Y RECOLECCION DE PRODUCTO Y
CON VENTANAS DE HORARIO NEGOCIABLES.**

Presentada por

JESÚS FABIÁN LÓPEZ PÉREZ

APROBADA POR LA COMISION DE TESIS

DR. CARLOS GÓMEZ DÍAZ DE LEÓN (PRESIDENTE)

**DR. MIGUEL ÁNGEL PALOMO GZZ
SECRETARIO**

**DR. GUSTAVO ALARCÓN MARTÍNEZ
PRIMER VOCAL**

**DR. JOSÉ LUIS ABREU QUINTERO
SEGUNDO VOCAL**

**DR. MOHAMMAD H. BADI ZABEH
TERCER VOCAL**

**DR. BERNARDO VILLARREAL CELESTINO
(PROFESOR VISITANTE DISTINGUIDO)**

San Nicolás de los Garza, Nuevo León, Mayo, 2004

Cualquier cosa que hagas será insignificante, pero es muy importante que tú lo hagas.

Mahatma Gandhi

I). PREFACIO DE LA TESIS.

La Ciencia de la Administración puede ser definida como un conjunto de conocimientos y técnicas que se utilizan para llevar a un organismo al logro de sus objetivos mediante la racionalización de sus recursos de operación. El racionalismo de Descartes y Kant manifiesta que las percepciones que recibimos de nuestro contacto con el mundo son inexactas. En ese sentido, las matemáticas a través de la historia, han jugado un papel importante en el desarrollo de la civilización dándole significado y organización. El presente trabajo está íntimamente asociado a la disciplina de la Investigación de Operaciones. No obstante su principal objetivo es cerrar la brecha entre la teoría de la optimización matemática versus su aplicación práctica en el contexto de la Ciencia de la Administración.

Tradicionalmente los problemas de optimización cuantitativa en el ámbito académico de la logística han sido atendidos mediante métodos de solución exacta. No obstante, la aplicación práctica de estos métodos para administrar la entrega y recolección de productos en empresas de distribución, frecuentemente ha sido rebasada por las limitantes de tiempo para la toma de decisiones del administrador del ruteo. Lo anterior origina la necesidad de investigar nuevas alternativas.

Dos de las revoluciones científicas más importantes del siglo XX fueron la invención de la computación electrónica y el desarrollo de las leyes que gobiernan los sistemas biológicos. Las Ciencias Computacionales, han sido utilizadas como medio para modelar y/o simular el proceso de evolución en los sistemas biológicos. A dicha modelación se le ha denominado “Algoritmos Genéticos”. Los Algoritmos Genéticos han sido vistos como una meta-heurística de optimización para problemas combinatorios, que aunque no asegura la obtención de la solución óptima exacta, si consigue obtener soluciones de buena calidad en tiempos de ejecución computacionales razonables para su aplicación en la práctica. Un algoritmo genético puede ser visto a través de tres binomios:

1. Selección + Mutación: éste binomio es el encargado de desarrollar el proceso de escalamiento u exploración del espacio solución alrededor de la solución local. Mientras el proceso de mutación genera variantes en el vecindario de la solución actual, el proceso de selección acepta solo aquellos cambios que tengan un mejor desempeño. Esto es lo que en términos humanos

diríamos, el proceso de mejora continua. Aunque este proceso de experimentación alrededor de la solución actual provee medios para el mejoramiento, no obstante carece de la innovación para inteligentemente saltar hacia lugares más prometedores del espacio solución.

2. Selección + Reproducción: éste binomio es el encargado de desarrollar en el algoritmo genético el proceso de innovación a diferencia del anterior relacionado a la mejora continua. Para ello, se parte del fundamento de que de la unión de características genéticas de dos organismos promedio, se puede conformar uno nuevo con la expectativa de que sea aún mejor. La idea primordial aquí tiene que ver con la identificación de aquellos bloques de material genético que convergen en las mejores soluciones. Así entonces, el proceso de innovación consiste en el intercambio apropiado de bloques genéticos.
3. Población + Selección: en éste binomio, diríamos que la verdad y la sabiduría recae en el “vox-populi”. La población no es solamente el origen de donde se obtienen los mejores individuos, sino también la prueba decisiva para legitimar cuando un organismo es óptimo.

A través del tiempo, la selección natural ha generado una amplia gama de formas vivientes las cuales están diseñadas para desempeñar óptimamente ciertas tareas especializadas. El éxito de la evolución natural, al menos en nuestro planeta, es innegable y provee evidencia suficiente de viabilidad para su aplicación en problemas de optimización matemáticamente intratables.

El propósito de la presente investigación es desarrollar y proponer un algoritmo genético que permita resolver de una manera útil para su aplicación en la práctica, el problema denominado “Ruteo de distribución para un vehículo con entrega y recolección de producto con restricciones de ventana de horario negociables (SPDP-sTW)”. Para comprobar la eficacia de la propuesta algorítmica se implementará un diseño experimental el cual hará uso de las soluciones óptimas obtenidas mediante un algoritmo de ramificación y corte sin límite de tiempo. Adicionalmente se incluye un breve análisis económico en lo referente al costo-beneficio que resulta de la implementación de la propuesta. Finalmente se añade un estudio predictivo mediante el cual se identifica la forma en la cual cada uno de los parámetros del algoritmo genético contribuyen a su eficiencia durante la exploración del espacio solución.

II). FILOSOFÍA DE LOS ALGORITMOS GENÉTICOS.

El principal tema en la filosofía de los algoritmos genéticos es en lo referente a la fuente de la inteligencia. La inteligencia existe en los cerebros de los seres vivos, en mayor o menor medida. No obstante, la inteligencia no existe sola, sino de manera asociada a la realidad a través de las interacciones con el medio ambiente. En esta interacción es que se desarrolla la evolución. Dicha interacción en la evolución de las sociedades humanas, se presenta a través de la inteligencia propia del individuo y así como en el nivel cultural del ambiente en donde se desarrolla. El ambiente cultural de una sociedad, intrínsecamente provee nuevas formas de inteligencia a sus integrantes.

Independientemente de la inteligencia personal con que genéticamente haya sido dotado un individuo, éste puede nacer con o sin este ambiente cultural a su favor. No obstante, es razonable pensar que la cultura ayudará, ya sea a positiva o negativamente, a establecer los paradigmas del individuo ante su medio ambiente. En otras palabras, en la descendencia de un individuo no es decisivo su material genético, sino que también existen aspectos tales como la herencia económica, la educación y los valores que trasmite a sus hijos. No hay duda de que este tipo de herencia no es genética, sino más bien del orden cultural, la cual privilegia a sus descendientes sobre el resto de la población. Así entonces es fácilmente verificable que generar una población a través de una herencia o conocimiento previo es mucho mejor que hacerlo tan solo a partir de la aleatoriedad. Hacerlo del último modo, equivaldría a renegar de la cultura heredada de las generaciones anteriores.

Con el contexto anterior, damos finalmente pie a lo que denominamos “La peligrosa idea de Darwin”. Esta consiste en establecer que el origen del ser humano se debe tan solo a un proceso mecánico de selección, reproducción y mutación en el cual no existió la necesidad de un conocimiento o herencia divina para explicar el origen del hombre. ¿Podemos hacer que la dignidad y religiosidad de la vida humana se mantenga intacta de frente al Darwinismo?. La respuesta no es sencilla y cae en el ámbito de la metafísica. No obstante, no podemos descartar la existencia de un Dios todopoderoso que con su herencia y sabiduría haya bendecido a las primeras poblaciones humanas y que por ende el proceso natural de la evolución haya logrado hacer el resto de una manera más eficiente debido a lo primero.

III). AGRADECIMIENTOS.

Una tesis como la presente, solo es posible a través del apoyo y motivación de la gente que hace posible el desarrollo de la ciencia. En ese preciso sentido mi más sensible agradecimiento al Dr. José Barragán Codina y al Dr. Mohammed Badii, ambos responsables del programa doctoral y quienes con su labor diaria, han sembrado la semilla de la ciencia en todos nosotros. A la Lic. María Eugenia García, de quién recibí personalmente su ayuda durante el proceso administrativo de la presente tesis. De manera muy especial al MA Jorge Castillo Villarreal, quién como Director de la Facultad, facilitó en todo momento los medios y recursos para llegar a la presentación del siguiente producto científico.

Finalmente, mi agradecimiento a la compañía Embotelladoras ARCA SA de CV por el apoyo económico que recibí de parte de ellos; gracias por la oportunidad que me brindaron para este proyecto personal en el cual yo siempre procuré retribuirles al ser un mejor profesional en el área de la logística.

IV). RECONOCIMIENTOS.

Mi reconocimiento a todos los doctores que participaron durante los seminarios de mi formación doctoral. La mente brillante de todos ellos siempre me dio los elementos necesarios para mantenerme en la dirección correcta.

Un reconocimiento especial y muy merecido a los miembros del comité de tesis por sus valiosas aportaciones hacia el proyecto de investigación; cada uno apropiadamente en su área de especialidad ha dado lo mejor de sí para enriquecer el producto científico que aquí yo presento también en nombre de ellos.

Finalmente, al Dr. Carlos Gómez Díaz de León, por ser mi tutor, mi amigo y haberme mostrado el camino hacia una filosofía de la administración en un sentido mucho más humano y trascendental.

V). DEDICATORIA.

Quiero dedicar esta tesis a quiénes han sido mi fuente de fortaleza y esperanza :

- ? A Dios, por haberme dado la oportunidad de conocer su más grande misterio y aún después, seguir creyendo en él con una fe renovada.
- ? A mis padres, Don Fabián López Laurel y Doña María Judith Pérez Muñoz, por todos sus sacrificios y haberme enseñado el valor de la honestidad y la disciplina.
- ? A mis hermanos, José Felipe, Judith y Ruth, por su paciencia y apoyo moral durante todos estos años. A ellos les digo que el esfuerzo y la persistencia vale la pena.
- ? Al padre de mi esposa, Don Pablo Méndez Martínez quién ha sido para mi un ejemplo de trabajo constante y ayuda al prójimo en su labor en el Hospital Universitario.
- ? A mis dos hijos, David y Alejandra, por todo el tiempo que no pude estar con ustedes. Quiero decirles que lo más importante que deben aprender en la vida, es el saber encontrar su felicidad.
- ? Y finalmente a mi esposa Betty, la fuente de todas mis inspiraciones. Gracias a ti, por mostrar siempre interés en todas mis ideas y pacientemente motivarme para llegar al fin de este proyecto. Después de haber ido y regresado a través este viaje intelectual, el mejor método que tengo para explicarte lo que encontré, es con la sencillez de nuestro amor.

No se como el mundo me verá a algún día, pero yo solo me siento como un niño que juega en la playa del mar y que jugando, encuentra de vez en cuando un guijarro mas liso o una concha más bonita de lo normal, mientras que el océano de la verdad yace desconocido ante mis ojos.

Isaac Newton

INDICE GENERAL.

Prefacio y reconocimientos.....	(i-viii)
Glosario de Términos.....	2
Lista de Tablas.....	3
Lista de Figuras.....	5
Tabla de Contenido.....	7
Primera parte: Antecedentes del proyecto.	
1. Introducción a la investigación de operaciones y a la logística.....	12
2. El factor educativo en la aplicación de la investigación de operaciones.....	33
Segunda parte: Marco conceptual.	
3. Exposición de problemas generales en la logística.....	56
4. Fundamentos matemáticos para problemas de logística de ruteo.....	96
5. Fundamentos del algoritmo genético como meta-heurística propuesta.....	124
Tercera parte: Desarrollo del proyecto de investigación.	
6. Planteamiento del problema de investigación... ..	158
7. Planteamiento de objetivos e hipótesis del proyecto de investigación.....	188
8. Diseño experimental.....	213
9. Resultados.....	246
10. Conclusiones.....	282
Anexos.....	288
Referencias Bibliográficas.....	323

Glosario de términos

APS: Advanced Planning Scheduling, Sistema de Planeación y Programación Avanzada

BC: Branch & Cut, Método de Ramificación y Corte

ERP: Enterprice Resource Planning, Sistema de Planeación de Recursos Empresariales

GA: Genetics Algorithms, Algoritmo Genético

INFORMS: International Federation of Operation Research and Mangament Science,

IP: Integer Programming, Programación Entera

LP: Linear Programming, Programación Lineal

MIP: Mixed Integer Programming, Programación Mixta Entera / Binaria

MRP: Materials Requirements Planning, Sistema para Planeación de Requerimientos de Materiales

MS: Management Science, Ciencia de la Administración

NLP: Non Linear Programming, Programación NO Lineal

NP-Hard: Non Polinomial Hard, Problema con tiempo de ejecución NO Polinomial

OR: Operations Research, Investigación de Operaciones

PDP-TW: Pickup & Delivery Problem with Time Windows,

Problema de entrega y recolección con ventana de horario

QAP: Quadratic Assignment Problem, Problema de Asignación Cuadrático

SA: Simulated Annealing, Enfriamiento Simulado

SCHEDULING: Secuenciación / Programación

SCM: Supply Chain Management, Administración de Cadena de Suministro

SOLVER: Optimizador

SPDP-sTW: Single pickup and delivery problem with soft time window

Problema de vehículo único con entrega y recolección y con ventana de horario negociable

TB: Tabú Search, Búsqueda Tabú

TSP: Traveling Salesman Problem, Problema del Agente Viajero

TSP-TW: Traveling Salesman Problem with Time Windows,

Problema del Agente Viajero con Ventana de Horario

VRP-TW: Vehicle Routing Problem with Time Windows,

Problema de Ruteo de Vehículos con Ventana de Horario

Lista de Tablas

Tabla 2.1 Niveles de importancia relativa.....	44
Tabla 2.2 Clasificación de la muestra de acuerdo al tipo de empresa donde labora el profesional.....	47
Tabla 2.3 Clasificación de acuerdo a la escuela de procedencia.....	47
Tabla 2.4 Clasificación de acuerdo a la carrera de procedencia.....	47
Tabla 2.5 Clasificación de acuerdo al año de graduación de la carrera profesional.....	48
Tabla 2.6 Clasificación de acuerdo al puesto ejercido... ..	48
Tabla 2.7 Clasificación de acuerdo al área de desempeño profesional... ..	48
Tabla 2.8 Incidencia porcentual de la aplicación de las matemáticas en la toma de decisiones.....	49
Tabla 2.9 Incidencia del grado de importancia atribuida a cada variable.....	49
Tabla 2.10 Incidencia del grado de importancia nivel “5” atribuido a cada variable.....	50
Tabla 2.11 Porcentajes de variabilidad explicada para cada variable.....	51
Tabla 2.12 Porcentajes de variabilidad explicada y confiabilidad para cada constructo.....	52
Tabla 2.13 Porcentajes de variabilidad explicada para las variables del constructo compuesto.....	53
Tablas 3.1 (a) y (b) Comparativa de software para optimización para aplicaciones de ruteo.....	87
Tablas 3.2 (a) y (b) Comparativa software de optimización para aplicaciones de ruteo (2da parte)... ..	88
Tablas 3.3 (a) y (b) Comparativa del software para planificación de rutas.....	90
Tabla 3.4 Distribución de empresas del AMM de acuerdo al sector industrial.....	91
Tabla 3.5 Estratificación de empresas de acuerdo al tipo de problema logístico.....	92
Tabla 3.6 Estratificación de empresas de acuerdo al tipo de solución que están llevando a cabo.....	92
Tabla 3.7 Estratificación de empresas que han implementado algún tipo de sistema ERP o APS.....	92
Tabla 3.8 Estratificación de empresas de acuerdo al tipo de secuenciación que requieren resolver.....	92
Tabla 4.1 Tiempo computacional de acuerdo al grado de complejidad de un problema.....	99
Tabla 4.2 Tabla esquemática de los métodos propuestos para la solución del TSP.....	120
Tabla 5.1 Ejemplificación del método de selección de la ruleta.....	135
Tabla 5.2 Cantidad de publicaciones hechas por año de Algoritmos Genéticos en la Logística.....	148
Tabla 5.3 Publicaciones por país de los Algoritmos Genéticos en la Logística respecto al total.....	148
Tabla 5.4 Publicaciones por Tema de Investigación de los Algoritmos Genéticos en la Logística... ..	149
Tabla 5.5 Publicaciones por Tema de Investigación de los Algoritmos Genéticos en la Logística....	149

Tabla 6.1 Resultados del experimento para la solución del ATSP-TW a través de BC.....	181
Tabla 6.2 Resumen comparativo de los estudios de investigación previos.....	183
Tabla 8.1 Matriz resultante del diseño del experimento.....	243
Tabla 8.2 Matriz de probabilidades para la prueba de la hipótesis.....	244
Tabla 9.1 Matriz de frecuencias para los 16 grupos experimentales.....	247
Tabla 9.2 (a) Matriz de valores calculados para el estadístico “T”.....	249
Tabla 9.2 (b) Matriz experimental de probabilidades “P Valor” para la prueba “T”.....	251
Tabla 9.3: Costos relacionados a la propuesta de solución basado en la aplicación de un Algoritmo Genético para solucionar el problema de ruteo SPDP-sTW en una empresa del AMM.....	255
Tabla 9.4: Ahorro potencial por ruta a ser disminuida de la flota vía eficiencia en el ruteo.....	256
Tabla 9.5: Análisis Costo-Beneficio derivado de la implementación del algoritmo propuesto.....	258
Tabla 9.6. Análisis comparativo de los beneficios obtenidos en términos de las distancias a recorrer entre una ruta de distribución actual versus la ruta optimizada propuesta por el algoritmo genético.....	260
Tabla 9.7: Diseño experimental para los parámetros N y G.....	266
Tabla 9.8: Diseño experimental para los parámetros FC y FM.....	267
Tabla 9.9: Diseño experimental para los parámetros N, G, FC y FM.....	267
Tabla 9.10: Análisis de Varianza Univariado: Prueba de efectos entre predictores.....	268
Tabla 9.11: Estimación estadística de los coeficientes e intervalos de confianza para los parámetros de operación del algoritmo genético.....	269
Tabla 9.12: Coeficientes obtenidos para el modelo polinomial.....	272
Tabla 9.13: Distribución estadística del costo en la función objetivo para el espacio solución.....	274

Lista de Figuras

Figura 1.1 Escuelas del pensamiento en la administración.....	12
Figura 1.2 Historial de los primeros 50 años del OR/MS.....	14
Figura 1.3 La decadencia del OR/MS.....	15
Figura 1.4 Efectos de las recomendaciones de INFORMS.....	15
Figura 1.5 El árbol de las técnicas del OR/MS.....	16
Figura 1.6 Mapa multidisciplinario del proyecto de investigación.....	18
Figura 1.7 Proceso básico para el ciclo de modelación.....	20
Figura 1.8 Metodología aplicada para el ciclo de modelación.....	25
Figura 1.9 Diagrama esquemático acerca de la relación entre el OR/MS y su aplicación en la logística..	32
Figura 2.1 Competencias buscadas en el OR/MS.....	37
Figura 2.2 Formato de encuesta para el estudio de campo para profesionistas de empresas del AMM...	43
Figura 2.3 Diagrama esquemático acerca del factor educación en la aplicación del OR/MS.....	54
Figura 3.1: Seis problemas generales en la práctica de la logística.....	56
Figura 3.2 Relación de los sistemas ERP con el OR/MS en el proceso de toma de decisiones.....	82
Figura 4.1 Intratabilidad matemática del TSP.....	100
Figura 4.2 Un tour en un TSP de ocho ciudades.....	101
Figura 4.3 Diagrama de flujo del algoritmo basado en la “búsqueda tabú”.....	114
Figura 4.4 Heurística del tipo “k2-opt” para el TSP.....	115
Figura 4.5: Ejemplo de un ruteo factible en un VRP.....	122
Figura 5.1 Acontecimientos históricos en el desarrollo de los algoritmos genéticos.....	126
Figura 5.2 Variantes del algoritmo genético.....	127
Figura 5.3 Fundamento del algoritmo genético.....	131
Figura 5.4 Desarrollo de la estructura cromosómica y política poblacional.....	134
Figura 5.5 Ruleta que representa los valores de aptitud de la tabla 5.1.....	136
Figura 5.6 Funcionamiento del operador de selección genético.....	137
Figura 5.7 Un solo punto de cruce. Cada pareja de organismos da origen a 2 descendientes.....	138
Figura 5.8 Dos puntos de cruce.....	138
Figura 5.9 Proceso de cruce genético y reproducción.....	140
Figura 5.10 Proceso de escalamiento para la búsqueda de la solución óptima.....	141
Figura 5.11 Afectación del factor de mutación y elitismo en el algoritmo genético.....	142

Figura 5.12 Pseudo-código del algoritmo genético simple.....	143
Figura 5.13 Funcionamiento del algoritmo genético.....	144
Figura 5.14 Relación entre la cantidad de aplicaciones de los Algoritmos Genéticos en el campo de la Logística con respecto al total.....	147
Figura 5.15 Ventajas en la aplicación de los algoritmos genéticos.....	153
Figura 7.1 Diagrama de flujo para la propuesta de solución del SPDP-sTW.....	210
Figura 8.1 Proceso de bifurcación en el procedimiento de ramificación y corte.....	217
Figura 8.2 Cálculo del tamaño de la muestra a partir de información disponible de las muestras.....	236
Figura 9.1: Distribución de frecuencias para la suma aritmética de parámetros del algoritmo genético.....	274
Figura 9.2: Distribución Acumulada del espectro del Espacio Solución del problema.....	275

Tabla de Contenido

Primera parte: Antecedentes del proyecto.

1. Introducción a la investigación de operaciones y a la logística.....	12
1.1 Introducción a la investigación de operaciones (OR/MS).....	12
1.2 Introducción a la modelación matemática.....	20
1.3 El enfoque evolutivo en el proceso de la modelación matemática.....	22
1.4 Etapas incluidas en el proceso de modelación matemática.....	24
1.5 Definición de la función logística.....	26
1.6 Aplicación del OR/MS en la práctica de la logística.....	28
1.7 Comentarios finales: El arte en la aplicación del OR/MS y el arte en la práctica de la logística.....	30
2. El factor educativo en la aplicación de la investigación de operaciones.....	33
2.1 Introducción: el proceso de enseñanza y aprendizaje en el OR/MS.....	33
2.2 La práctica profesional del OR/MS en las empresas.....	35
2.3 Las competencias requeridas para la aplicación del OR/MS.....	37
2.4 El desarrollo de competencias en México.....	40
2.5 Caso de estudio: La aplicación de las matemáticas por parte de los profesionistas de las empresas del área metropolitana de la ciudad de Monterrey.....	42
2.6 Comentarios finales de la problemática educativa en la aplicación del OR/MS.....	54

Segunda parte: Marco conceptual.

3. Exposición de problemas generales en la logística.....	56
3.1 Introducción.....	56
3.2 Primer problema de la logística: planeación estratégica de red e infraestructura.....	57
3.3 Segundo problema de la logística: programación de la producción para ambientes continuos..	67
3.4 Tercer problema de la logística: programación de la producción para ambientes discretos.....	74
3.5 Cuarto problema de la logística: programación de la distribución (Ruteo).....	80
3.6 La relación de los sistemas MRP, ERP y APS con el OR/MS.....	82
3.7 Identificación y análisis comparativo respecto a las alternativas de software comercial tipo APS disponibles en el mercado para problemas de distribución de ruteo.....	86
3.8 Caso de estudio: revisión de las estrategias de solución implementadas en la ciudad de Monterrey para el apoyo en la toma de decisiones de la función logística.....	91
3.9 Comentarios finales.....	95
4. Fundamentos matemáticos para problemas de logística de ruteo.....	96
4.1 Introducción: significado del concepto “problema difícil de resolver” (NP-Hard).....	96
4.2 Introducción al problema del agente viajero (TSP).....	101
4.3 Aplicación del TSP en problemas generales de secuenciamiento en manufactura.....	103
4.4 Clasificación de los métodos propuestos para solución del TSP.....	105
4.5 Revisión de métodos basados en heurísticas para la solución del TSP.....	107
4.6 Revisión de la meta-heurística “tabu-search” para la solución del TSP	111
4.7 Revisión del método basado en programación dinámica para la solución del TSP.....	117
4.8 Revisión de métodos basados en ramificación y corte para la solución del TSP.....	118
4.9 Comentarios finales: resumen esquemático y otras variantes del TSP	120

5. Fundamentos del algoritmo genético como meta-heurística propuesta.....	124
5.1 Introducción: breve historia del antecedente de los algoritmos genéticos.....	124
5.2 La versión americana y la versión alemana del algoritmo genético.....	125
5.3 Introducción a los parámetros de operación del algoritmo genético.....	128
5.4 La población de organismos y la función de desempeño.....	132
5.5 La definición del cromosoma del organismo y sus genes.....	133
5.6 El operador de selección.....	135
5.7 El operador de cruce o reproducción.....	138
5.8 El Operador de mutación.....	141
5.9 Funcionamiento general del algoritmo genético simple.....	143
5.10 El problema de la codificación de la solución (genoma) en los algoritmos genéticos.....	145
5.11 Trabajos previos de investigación: aplicación de los algoritmos genéticos para optimización en problemas de logística.....	147
5.12 Ventajas y desventajas del algoritmo genético como estrategia de solución.....	153
5.13 Variantes adicionales en los algoritmos genéticos.....	155
5.14 Comentarios finales.....	157

Tercera parte: Desarrollo del proyecto de investigación.

6. Planteamiento del problema de investigación.....	158
6.1 Introducción al problema de investigación.....	158
6.2 Planteamiento formal del problema de investigación:	
Ruteo de distribución para un vehículo con entrega y recolección de producto con restricciones de ventana de horario negociables (SPDP-sTW).....	161
6.3 Modelación matemática del problema de investigación.....	163
6.4 Aproximaciones para la solución del SPDP-sTW vía algoritmos basados en el TSP.....	170
6.5 Aproximaciones para la solución del SPDP-sTW basadas en el TSP-TW y en VRP-TW....	173
6.6 Revisión algorítmica referente a trabajos de investigación previos al problema PDP-TW.....	175
6.7 Identificación del estado del arte para el problema de investigación.....	179
6.8 Comentarios finales: delimitación y diferenciación de la variante del problema de investigación.....	184
7. Planteamiento de objetivos e hipótesis del proyecto de investigación.....	188
7.1 Introducción.....	188
7.2 Planteamiento del objetivo central del proyecto de investigación.....	190
7.3 Planteamiento de los objetivos específicos y entregables del proyecto de investigación.....	191
7.4 Delimitación del proyecto de investigación.....	192
7.5 Justificación científica del problema de investigación.....	194
7.6 Relevancia práctica del problema de investigación.....	197
7.7 Propuesta metodológica para la solución del problema de investigación.....	198
7.8 Declaración de la hipótesis de trabajo del proyecto de investigación.....	211
7.9 Comentarios finales: taxonomía general del proyecto de investigación.....	212
8. Diseño experimental.....	213
8.1 Introducción: objetivos del esquema general del experimento.....	213
8.2 Aplicación del procedimiento algorítmico de ramificación y corte (BC) como grupo control para la prueba experimental.....	216
8.3 Técnica estadística para la prueba de la hipótesis de investigación.....	222
8.4 Terminología empleada en el diseño de experimentos.....	224

8.5 Metodología empleada para la planificación del experimento.....	226
8.6 Estrategia para la asignación de las unidades experimentales en el experimento.....	229
8.7 Posibles errores en el diseño de experimentos.....	230
8.8 Definición de los parámetros a ser manipulados en el experimento.....	231
8.9 Medición de las variables para la prueba experimental.....	232
8.10 Estudio descriptivo de las instancias numéricas a ser utilizadas en el experimento.....	233
8.11 Cálculo para el tamaño de la muestra.....	235
8.12 Planteamiento de la prueba experimental.....	238
8.13 Implementación del experimento.....	241
9. Resultados.....	246
9.1 Introducción.....	246
9.2 Prueba de la hipótesis de investigación.....	248
9.3 Análisis costo-beneficio para la aplicación del Algoritmo Genético propuesto	254
9.4 Estudio analítico para los parámetros de operación del Algoritmo Genético.....	263
9.5 Diseño experimental para la predicción de los parámetros del Algoritmo Genético.....	266
9.6 Significancia de los parámetros del Algoritmo Genético en relación a su eficacia.....	276
9.7 Comentarios finales.....	281
10. Conclusiones.....	282
11. Anexos.....	288
i. Anexo A: Fase de descomposición basada en el método SPP para el análisis y explotación topológica de la versión original de la red logística	288
ii. Anexo B: Fase de compresión vía estrategia de clusterización mediante criterios de maximización de afinidad	291
iii. Anexo C: Fase de compresión discriminante de arcos basada en la heurística de los “k” nodos vecinos más cercanos para obtener la versión compacta de la red	294
iv. Anexo D: Fase de generación agresiva de cortes aplicada a la versión compacta de la red.....	297
v. Anexo E: Fase evolutiva generacional para explotar el “pool” de cortes generados	313
vi. Anexo F: Fase de descompresión de la ruta propuesta para la versión original de la red.....	319
12. Referencias Bibliográficas.....	323

El hombre intenta crearse para sí mismo un dibujo simplificado e inteligente del mundo; luego entonces él busca sustituir esta creación suya en lugar de la experiencia del mundo, y por ende la sobrepasa.

Albert Einstein

1. Introducción a la investigación de operaciones y a la logística.

1.1 Introducción a la investigación de operaciones (OR/MS).

Las Matemáticas desde una perspectiva histórica, nacen de la naturalidad y de la evolución del lenguaje como un medio de comunicación poderoso, conciso y sin ambigüedades. No se trata de un objeto de conocimientos ya constituido sino de una forma abierta a la creatividad y al razonamiento. Aunque debemos reconocer que la teoría de la administración tiene su origen a finales del siglo XIX, la aproximación que nosotros abordaremos se sitúa a principios de los años 40 del siglo XX. En este momento en el tiempo inicia una nueva escuela del pensamiento de la administración cuyo fundamento presupone la aplicación de los modelos matemáticos como medio de solución a los problemas, dando origen de este modo a la escuela del pensamiento denominado "Cuantitativista"¹. Esta corriente filosófica se concibe como el origen de la "Investigación de Operaciones" o también denominada "Ciencia de la Administración" (Operations Research & Management Science, OR/MS de sus siglas en inglés). Como reseña de esta evolución y con la finalidad de ubicar a la escuela "Cuantitativista" de la Administración en el contexto histórico a continuación en la Figura 1.1 se enumeran junto con sus precursores.

1. Administración científica. Iniciadores: Taylor, Emerson, Hamilton, Pratt, Gilbreth.
2. Gerencia científica. Sus precursores: Fayol, Gulick, Urwick, Koontz, Newman.
3. De la Burocracia. Iniciada por Max Weber.
4. De las Disfunciones burocráticas. Son iniciadores: Merton, Selznick, Downs, Blau, Scott.
5. De las Relaciones humanas, con sus precursores: Mayo, Dubin, Brown.
6. De la Gerencia interactiva. Expositores: Follet, Barnard, Drucker, Mintzberg.
7. De los Recursos humanos. Con autores: McGregor, Likert, Argyris, Maslow.
8. La Cuantitativista. Cuyos precursores son: Ackoff, Simon, Vroom.
9. La de Sistemas, con precursores: Von Bertalanffy, Churchman, Emery, Katz.
10. De Desarrollo organizacional, con autores: Margulies, Raia, French, Bennis, Dyer.
11. De la Cultura organizacional, con autores: Morgan, Smircich, Peter, Waterman, Meyer.
12. Del Control de calidad, con autores como: Deming, Feigenbaum, Ishikawa, Crosby.

Figura 1.1 Escuelas del pensamiento en la administración².

¹ Pawda, Juan. (2000), Modelos de Investigación de Operaciones, Vol 1, 1era.Ed, Limusa, México, pag 20-23.

² Rodríguez, Joaquín. (1998), Introd. a la Administración con Enfoque de Sistemas, 3era Ed, ECAFSa, México, pag 67.

El OR/MS tiene su origen en la Gran Bretaña en el año de 1939. En la Universidad de Manchester se formó un grupo interdisciplinario de individuos cuyo propósito fue estudiar el diseño óptimo de un sistema de defensa antiaéreo en el marco de la segunda guerra mundial. Este grupo estaría formado por psicólogos, físicos, astrofísicos, militares, topógrafos y matemáticos. Más adelante otros países tales como Estados Unidos, Canadá y Francia replicaron este tipo de grupos interdisciplinarios antes descrito.

En la década de los 50's ya terminada la guerra, en la Gran Bretaña se empezó a propagar la aplicación del OR/MS hacia el sector industrial. No obstante, los Estados Unidos durante esa época, enfocan y mantienen su aplicación exclusivamente en el campo militar. Quizás la razón de lo anterior tendrá que ver con el arraigo que el sector manufacturero en los Estados Unidos tenía en el uso de la Ingeniería Industrial como área de conocimiento para la solución de problemas que tuvieran que ver con la optimización de los procesos.

Fue hasta la década de los 60's con el desarrollo de la tecnología computacional, cuando el OR/MS se afianza en el mundo industrializado. La disciplina empieza a ser objeto formal de estudio en varias universidades del mundo tales como: University College of London, M.I.T, Case Western, Johns Hopkins, North-Western, Imperial College y la London School of Economics. Desafortunadamente, vendría más adelante en la década de los 70's un relajamiento de la disciplina en el contexto académico. A esta le seguiría en la década de los 80's una decadencia propagada ahora hacia su aplicación en el ámbito profesional³.

Con la finalidad de ilustrar la reseña histórica del OR/MS a continuación en la Figura 1.2 se muestran los principales acontecimientos ocurridos desde 1940 hasta 1980, con lo cual se cubren los primeros 40 años de la disciplina.

³ Geoffrion, A rthur. (1999), The decadence of operations research, ORMS Today, EJA, pag 3.

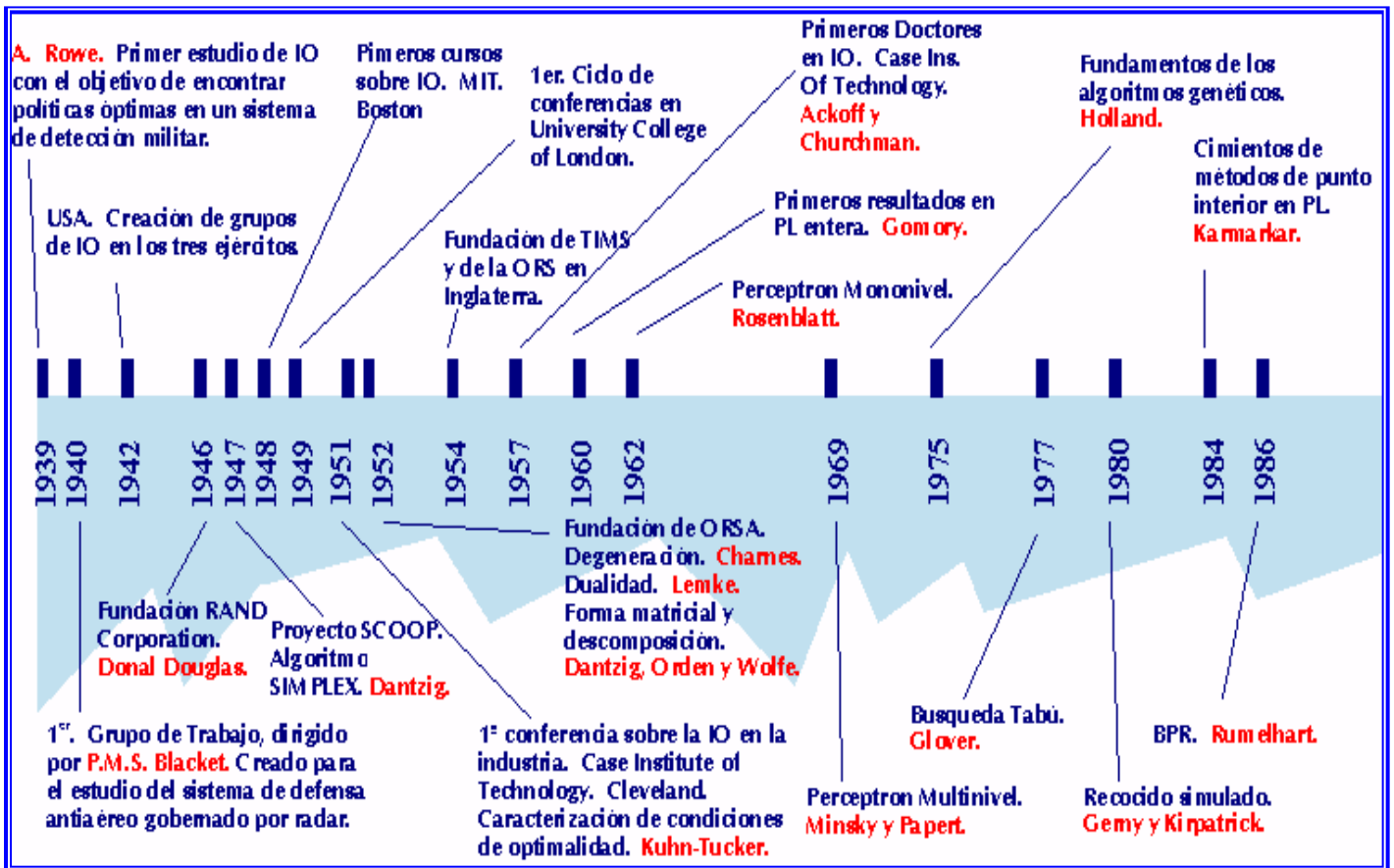


Figura 1.2 Historial de los primeros 50 años del OR/MS⁴

No fue sino hasta 1991, cuando se presenta el punto más crítico en el historial del OR/MS, año en el que el AACSB (American Assembly of Collegiate Schools of Business) quita la obligatoriedad de la materia en los programas de postgrado en Administración de Negocios en las Universidades de los Estados Unidos. La respuesta tardía a esta situación se da hasta el año 1996 cuando la organización INFORMS (International Federation of Operation Research and Mangament Science), designa una comisión con el propósito de investigar la problemática y desarrollar una autocrítica y un replanteamiento hacia el interior de la disciplina desde un punto de vista tanto académico como profesional. Las Figuras 1.3 y 1.4 a continuación resumen los aspectos más relevantes referidos a la época que se ha nombrado como de decadencia del OR/MS la cual se gesta en la década de los 70's y se replantea positivamente en 1996 a través de INFORMS.

⁴ Pawda, Juan. (2000), Modelos de Investigación de Operaciones, Vol 1, 1era.Ed, Limusa, México, pag 73.

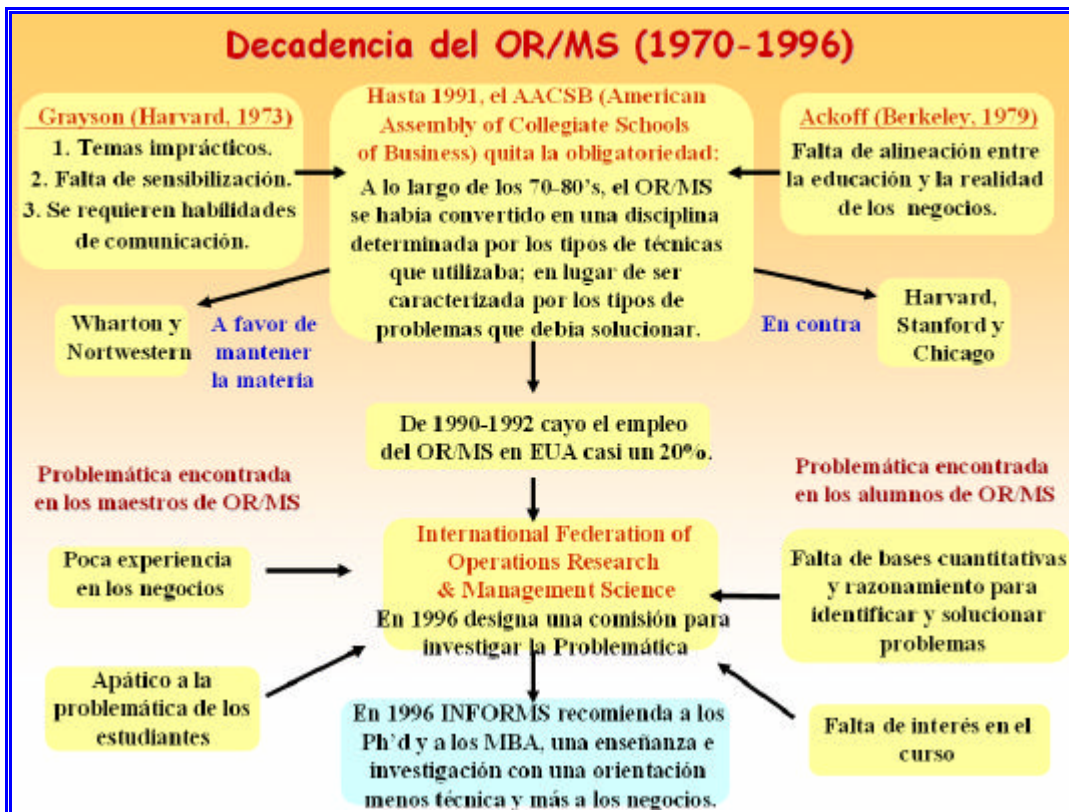


Figura 1.3 La decadencia del OR/MS.

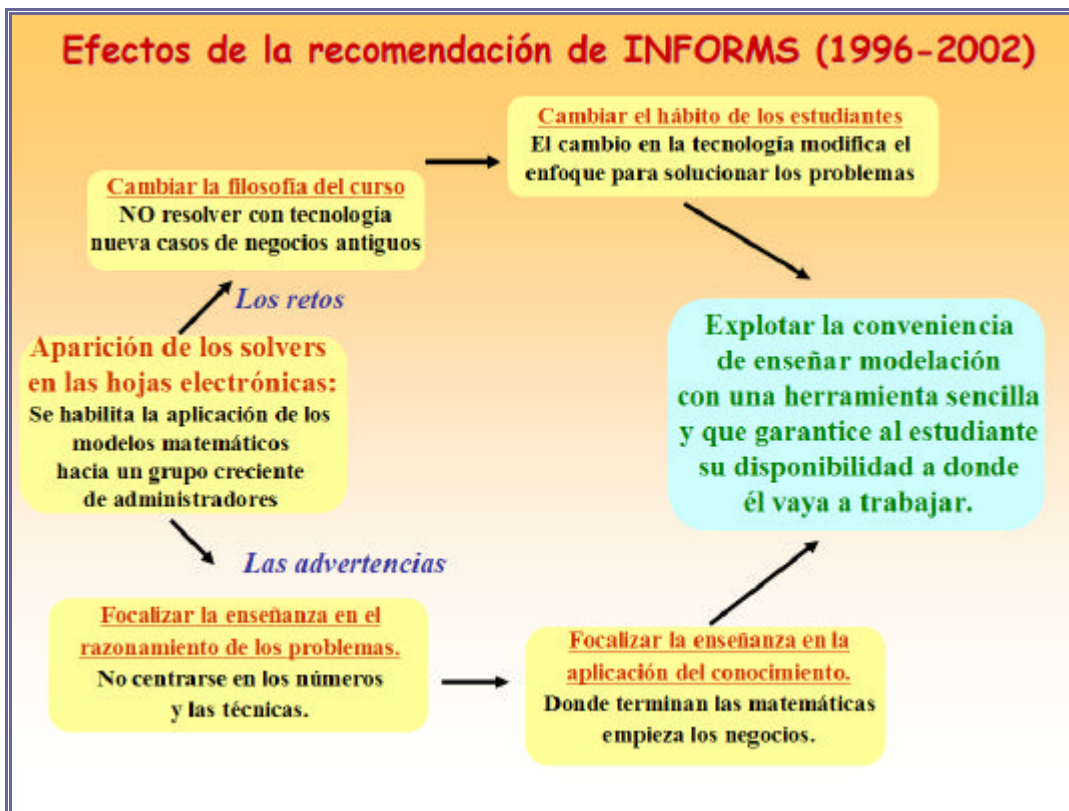


Figura 1.4 Efectos de las recomendaciones de INFORMS.

Hasta aquí una pequeña reseña histórica de los primeros 60 años del OR/MS. Para finalizar este aspecto del tema creemos que es importante que se visualice de manera general el cúmulo de técnicas que se encuentran aglutinadas en la disciplina del OR/MS. Por tal motivo, en la figura 1.5 se exponen a manera de un árbol todas ellas, con la finalidad de observar no solamente la gran diversidad de éstas, sino también para revisar las precedencias que se han venido dando entre cada una de ellas en un sentido histórico.

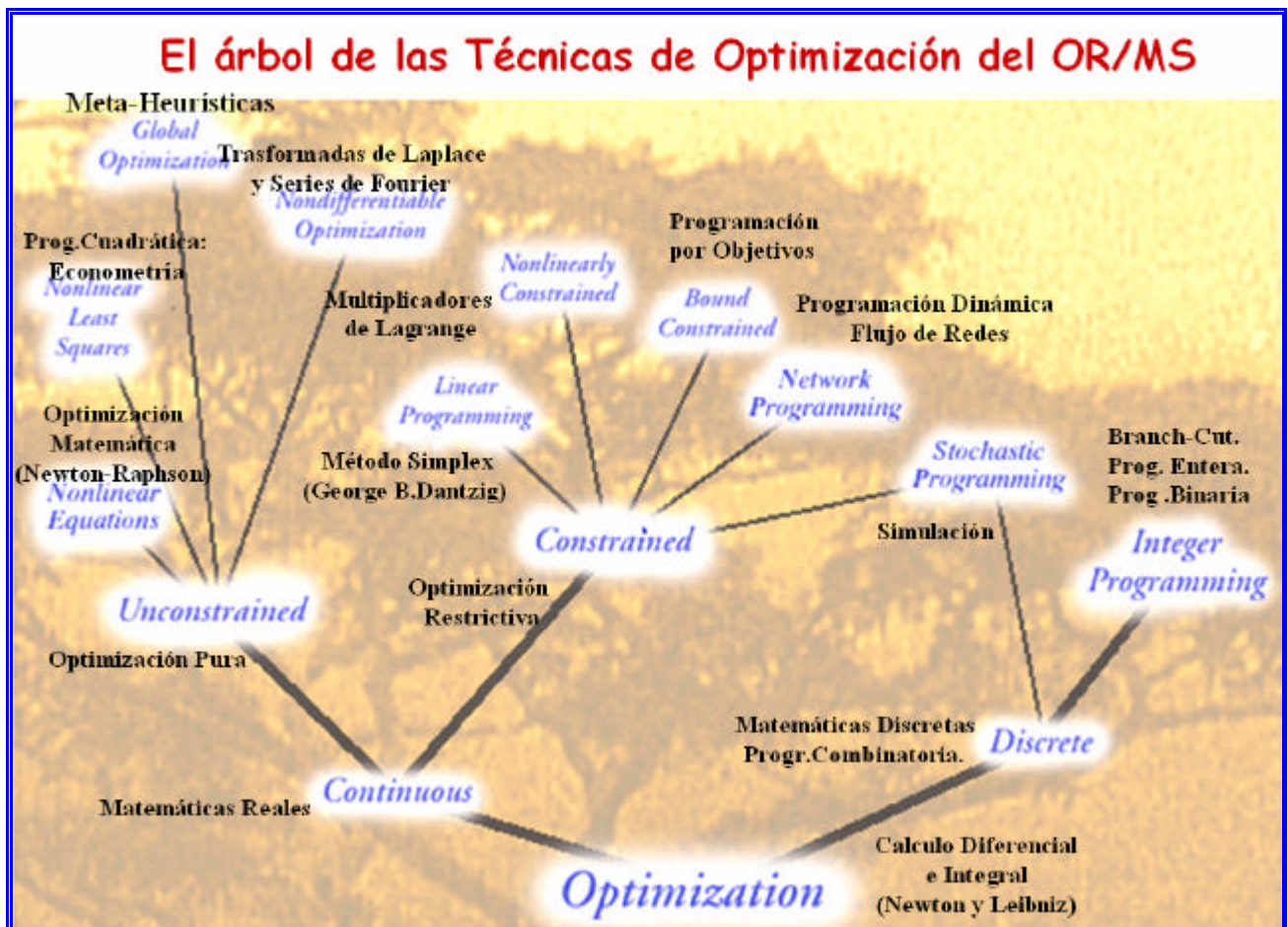


Figura 1.5 El árbol de las técnicas del OR/MS⁵.

⁵ Mittelman, H. (2002), Decision Tree for Optimization Software, disponible en <http://plato.la.asu.edu/guide.html>.

Partiendo del acrónimo "OR/MS", podemos en un sentido riguroso diferenciar que "MS" (Ciencia de la Administración) dedica más de su espacio al razonamiento y al planteamiento abstracto de los problemas, mientras que "OR" (Investigación de Operaciones) se concentra mayormente en las técnicas de solución matemáticas y computacionales a ser aplicadas para resolver el modelo abstracto previamente desarrollado. No obstante, en la actualidad ambas disciplinas diferenciadas se han conjuntado en un solo cuerpo de conocimientos. Lo anterior se hizo necesario, ya que ambas disciplinas se requieren mutuamente. La primera se enfoca en la modelación y la segunda en desarrollar las técnicas necesarias para lograr resolver numéricamente los modelos creados en la primera.

El OR/MS se define en términos generales por el carácter mensurable de sus objetos así por la forma matemática y lógica de sus razonamientos.⁶ Podemos definir al OR/MS como la aplicación de las matemáticas, la lógica y del método científico para la solución de problemas multidisciplinarios relacionados con la optimización de los procesos operativos en general. Para esto se apoya en la modelación matemática la cual intrínsecamente incluye la comprensión de los procesos a ser optimizados. Un modelo matemático, es aquella forma superior de la creatividad que tiene su origen en el dominio del lenguaje matemático. El lenguaje matemático es el resultado histórico de buscar simplificar la realidad, requiriendo en cada caso que el individuo desarrolle sus propias significaciones y relaciones.

El OR/MS tiene la finalidad de explicar los fenómenos basándose en la racionalidad establecida a través de las relaciones entre los objetos. Estas relaciones definidas entre las variables y sus límites o dominios de operación, permiten establecer espacios de solución finitos durante los procesos de búsqueda y de optimización matemática. La importancia del OR/MS está en que la competencia se ha vuelto cada vez más global y hoy más que nunca los administradores están obligados a buscar nuevas aproximaciones teóricas y metodológicas para hacer frente a la competitividad.⁷

La presente tesis está orientada hacia la aplicación del OR/MS para la solución de un problema específico de Programación de Ruteo en el ámbito de la logística. Con la finalidad de contextualizar la relación entre el área problemática referente (Logística) y el área de solución propuesta (OR/MS), a continuación se presenta la Figura 1.6 que muestra la naturaleza multidisciplinaria del proyecto:

⁶ Moore, Lawrence. (1993), Management Science, 4thEd, Allyn and Bacon, EUA, pag 3.

⁷ Bryne, John. (1997), Los Nuevos gurús de la Administración, "Revista Gestión y Estrategia", No.11-12, México.

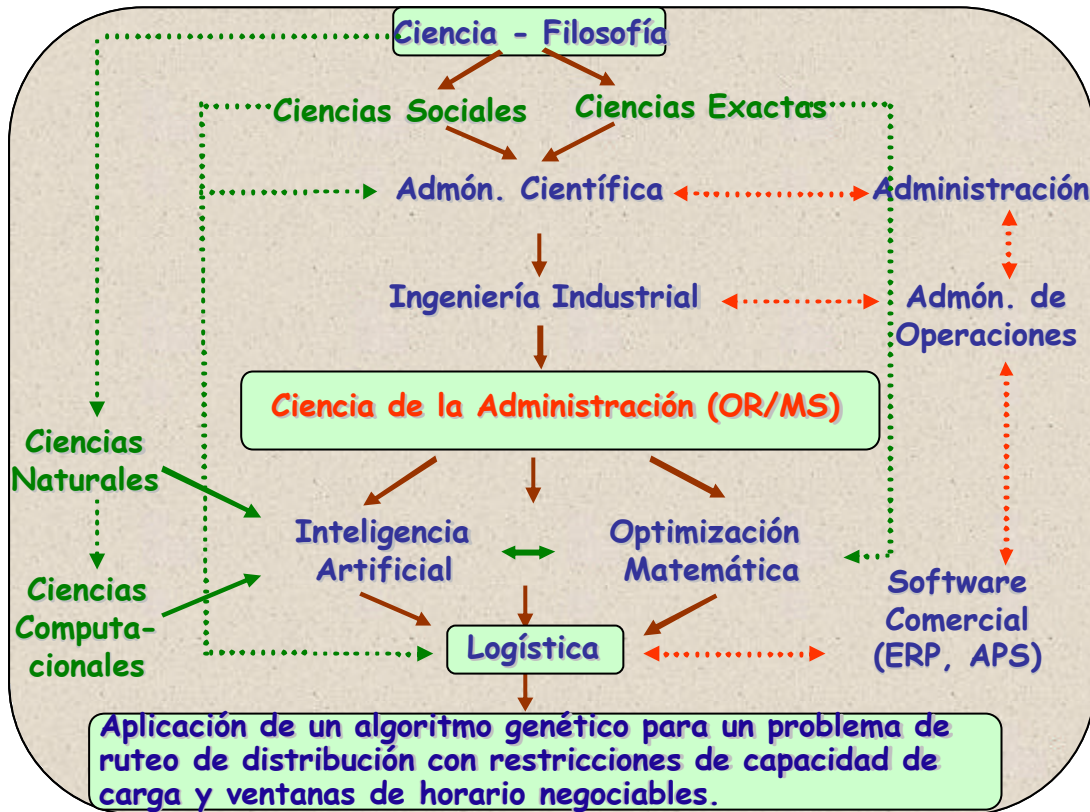


Figura 1.6 Mapa multidisciplinario del proyecto de investigación.

De la figura 1.6, podemos resumir que nuestro problema de investigación proviene del área de aplicación de la logística. La problemática de la logística a su vez actualmente está siendo atendida por dos habilitadores: (1) el OR/MS y (2) el software computacional basado en sistemas ERP (Enterprise Resource Planning) y en sistemas APS (Advanced Planning Scheduling). La herencia del primer habilitador proviene primordialmente de la Administración Científica y en el caso del segundo se orienta más a la Administración de Operaciones. Aunque nuestro problema parte de una circunstancia orientada a la actividad humana (ciencias sociales), nuestra propuesta de investigación, hará uso de diversas ramas de las ciencias, tales como las ciencias exactas, las ciencias naturales y las ciencias computacionales. Básicamente la combinación de las dos últimas ciencias mencionadas, brinda el fundamento a lo que denominamos, “algoritmo genético” que será propuesto en el capítulo # 5.

El lenguaje natural es insuficiente para responder a las necesidades de comunicación en el campo científico, por tal necesitamos un lenguaje apoyado en signos y códigos. La evolución del lenguaje natural a través de abstracciones ha contribuido convenientemente a mantener la información referencial, conceptual y lógica de los problemas que se requieren resolver. Esto último es lo que da origen al concepto de Modelación Matemática que será tratado a continuación.

El OR/MS, por supuesto, no solo contiene un especial interés por su aplicación en las áreas de logística. Existe una amplia y diversa lista de áreas de aplicación. Tomando en cuenta que nuestro proyecto de investigación de aquí en adelante se enfocará hacia su aplicación para un problema particular del área de logística, hemos considerado apropiado incluir una breve lista de áreas de aplicación mediante la cual resaltamos su importancia como área de conocimiento y de aplicación. A continuación se expone para dicho propósito la siguiente Tabla 1.1

Tabla 1.1 Áreas de aplicación generales del OR/MS ⁸.

Econometría y pronósticos
Ingeniería financiera
Planeación de presupuestos e inversiones
Planeación de recursos humanos
Programación de actividades
Planeación de transporte y tráfico
Planeación de instalaciones y layouts
Planeación de vida útil y reemplazos
Administración de inventarios
Administración de proyectos
Administración de servicio al cliente
Administración de manufactura y operaciones
Planeación de seguros
Planeación de Mercadotecnia
Ingeniería Industrial

⁸ Arsham, Hossein. (1996), Applied Management Science, Cambridge University Press, EUA, pag 282-289.

1.2 Introducción a la modelación matemática.

El hombre busca el conocimiento en virtud de que las cosas son susceptibles a ser determinadas y así de esta forma transforma el mundo en algo cognoscible. En la medida en la cual la realidad se transforma dinámicamente, es decir, dejan de existir algunas cosas sin resolver y resultan luego otras por resolver, en esa misma dinámica de temporalidad es que se desarrolla el conocimiento.

El proceso de construcción de un modelo de OR/MS proviene de aquel instinto humano para la creación de una representación abstracta y simplificada de la realidad con la finalidad de obtener un mejor entendimiento a un problema. El propósito de la modelación es la comprensión de los problemas y no tanto su solución computacional. Los modelos de OR/MS son descriptivos ya que muestran de manera abstracta la realidad incluyendo aspectos que de manera evidente o intuitiva sirvan para comprender los fenómenos que ocurren en dicha realidad y no solamente la observación de los síntomas superficiales. En la Figura 1.7 se muestra el proceso básico referente a la modelación cuyo objetivo es apoyar la comprensión y solución de los problemas.

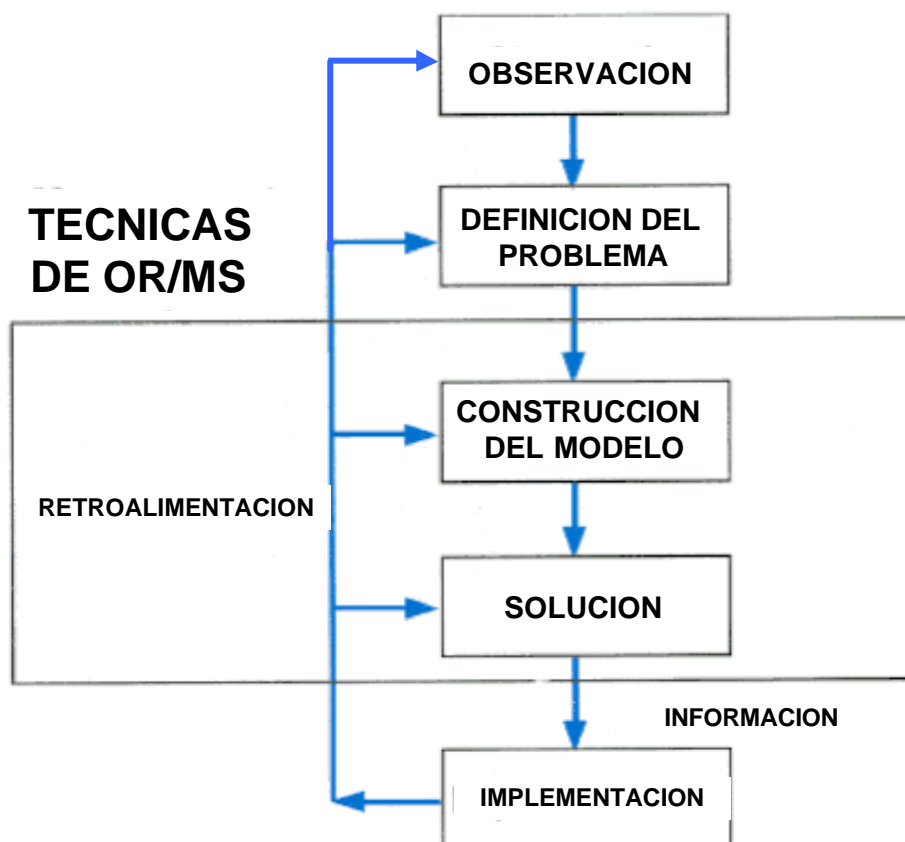


Figura 1.7 Proceso básico para el ciclo de modelación

Así pues, la comprensión de los problemas en los negocios, le ofrece al administrador la posibilidad de orientar su esfuerzo de manera práctica en la forma de cómo eliminar aquellas restricciones que estén limitando al sistema global. Por esa razón, el administrador debe tener las habilidades necesarias para poder analizar las implicaciones de sus modelos conceptuales. Es la modelación matemática, la que permite a los individuos explicar las relaciones lógicas que buscan dar respuesta a los objetos a ser estudiados. En la modelación matemática, cuando hablamos de resolver un problema, nos referimos a encontrar aquel conjunto de datos numéricos que cumplen las restricciones impuestas en el problema y que a la vez alcanzan un desempeño óptimo respecto a un objetivo previamente establecido.

Los modelos matemáticos sirven de guía en el proceso de toma de decisiones. Un buen modelo de OR/MS es aquel que es capaz de predecir el comportamiento del objeto de estudio de tal manera que se esté en posibilidad de encontrar su configuración de operación óptima. Su dificultad tiene que ver con el desarrollo de las simplificaciones que se deban hacerse de las variables involucradas en el modelo así como de sus relaciones. Esto resulta crucial para la viabilidad del OR/MS en la logística en virtud de que las soluciones generadas puedan ser implementadas en la práctica. En la modelación matemática, es mucho más importante la comprensión y la modelación de los procesos que lo relacionado a las técnicas matemáticas para la solución de los modelos.⁹

Por tanto es de esperar que el administrador deba estar más preocupado en hacer las averiguaciones relevantes para asegurar el entendimiento de un problema más que en el desarrollo de una solución matemática. Dicho de otro modo, a un administrador moderno debe estar más orientado a la definición de posibles alternativas de decisión, que a reducir su ámbito de acción en solo la determinación de la alternativa matemáticamente óptima.

⁹ Baker, Kenneth. (1999), Gaining Insight in Linear Programming, "Informs", Vol.1, EUA, Pag 4.

1.3 El enfoque evolutivo en el proceso de la modelación matemática.

Una forma de analizar el desarrollo de los modelos de OR/MS en cualquier ámbito de aplicación (por ejemplo en las áreas de logística), puede ser visto como un proceso evolutivo en el cual se inicia con un modelo muy simple pero también muy alejado de la realidad y entonces se continúa el proceso mediante el cual el modelo se va haciendo más elaborado y va dando a conocer la complejidad de la problemática logística. En efecto, el desarrollo de un modelo de OR/MS puede facilitarse iniciándolo a partir de estructuras lógicas, que utilizadas de manera general, sirvan como puntos de arranque para la sofisticación de modelos más elaborados. Esto equivale al concepto de "portabilidad" y "escalabilidad".

Las analogías o asociaciones de los modelos previamente desarrollados y probados, juegan un papel crítico en la determinación del punto de arranque del proceso evolutivo aplicado al desarrollo de los modelos. Aquí existe el concepto de "re-uso" de las estructuras lógicas de los modelos para ser aplicados en situaciones diversas. En el desarrollo de los modelos, existe la posibilidad de dividir los grandes problemas en instancias más pequeñas e independientes logrando con esto simplificar su tratabilidad y mantenibilidad de los modelos que aunque parecen resultar grandes a simple vista, en realidad tienen una dimensionalidad sub-dividible en función a la independencia de sus sub-problemas componentes.

Por otro lado, la parametrización del modelo matemático debe de mantenerse simple, tratando de capturar aquellos aspectos que realmente son relevantes y relegando o difiriendo aquellos otros que no lo sean tanto; sólo hasta aquel punto en el cual el usuario o el administrador demuestren que es relevante considerar dicho aspecto en el modelo y que además están preparados para continuar el proceso de sofisticación y evolución del modelo matemático.¹⁰ El desarrollo del modelo se logra mediante el uso de un simbolismo apropiado que pueda cubrir los aspectos de definición y dominio de las variables, lógica relacional y dimensionalidad del problema a resolver. Los símbolos deben ser sugestivos y sin ambigüedades con la finalidad de poder a través de ellos dar lectura a la abstracción de la realidad.

El modelo matemático también debe ser adaptativo al entorno a medida que nueva información pertinente y relevante vaya apareciendo. Esto se cubre en parte a través de la adecuada parametrización en el modelo ya antes mencionada. La parametrización por ejemplo, permite el manejo dinámico de los datos entrantes al modelo, los cuales pueden conformar diversas instancias cambiantes para un mismo problema. Pero el

¹⁰ Berry, William. (1990), Management Decision Science, 1thEd, Irwin, EUA, pag 563.

modelo debe ser adaptativo no sólo en el tratamiento de los datos sino también debe estar preparado para posibles modificaciones en su estructura lógica. Esto significa en que no tenga defectos de diseño estructural que le impidan continuar su proceso evolutivo y que permitan llegar a un modelo que, de manera completa, más no necesariamente compleja, incluya fielmente los fenómenos que mayormente describen la realidad de los procesos logísticos.

Finalmente, en cada una de las etapas durante la evolución de un modelo de OR/MS; y en aquellos casos en los cuales la nueva instancia del modelo ya no permita su tratabilidad en la práctica; existirá entonces la posibilidad de poder retroceder hacia una versión anterior de su evolución en búsqueda del refugio y la purificación que la simplicidad le pueda brindar al mantener el modelo implementable.¹¹

¹¹ Berry, William. (1990), Management Decision Science, 1thEd, Irwin, EUA, pag 494.

1.4 Etapas incluidas en el proceso de modelación matemática.

El proceso evolutivo y de enriquecimiento de los modelos de OR/MS convenientemente debe iniciar con una buena estrategia para promover la idea acerca de lo que es posible mejorar a partir del modelo conceptual del administrador. Asumiendo que todo modelo es perfectible por ser sólo una abstracción estática de una realidad siempre cambiante, es entonces posible reconocer las limitaciones de un modelo abstracto para primero representar la realidad y luego optimizarla. Lo anterior es lo que finalmente da origen a la naturaleza iterativa y evolutiva del proceso de modelación matemática.

Este proceso de enriquecimiento y escalabilidad en los modelos de OR/MS, involucra la necesidad de probar de manera rigurosa cada una de sus etapas evolutivas. No se puede asumir que aspectos esenciales o primitivos de un problema podrán ser, mejor o al menos igualmente tratados, por versiones más evolucionadas del modelo original. Aquí, el método científico se aplica en este ir y venir durante el proceso evolutivo de los modelos; ya que al no tratarse de un solo modelo, tampoco deberá ser verificada su eficiencia a través de la evidencia de una sola prueba experimental.

Para esto último, será necesario utilizar datos numéricos entrantes al modelo los cuales conformarán una instancia numérica para el modelo matemático en cuestión. Una instancia particular para el modelo representa una fotografía que fielmente puede dar a conocer aspectos como la complejidad y la dimensionalidad del problema a ser resuelto. Adicionalmente, una instancia numérica permite visualizar supuestos explícitos y algunas otras generalidades que hasta antes no aparecían evidentes en la definición previa del modelo. Las instancias numéricas por igual permiten determinar paramétricamente cuales supuestos deben ser mayormente tomados en cuenta y cuáles otros pueden ser relajados.

Partiendo del nivel de significación que cada persona tenga de un objeto y de sus relaciones, la comprensión del conocimiento difícilmente puede ser transferido a otras personas resultando en una equivalencia o copia del mismo. Así pues, debemos advertir que la habilidad creativa necesaria para el proceso de modelación, no será mayormente obtenida a través del proceso de identificación de los datos relevantes para un problema sino más bien a través de la experiencia que sólo se alcanza mediante el recorrido del proceso evolutivo que se inicia a partir de modelos sencillos para luego enriquecerlos gradualmente con todos aquellos supuestos que acerquen por aproximación al modelo hacia la realidad del proceso logístico pero manteniendo siempre la tratabilidad del modelo.

El proceso de modelación en el OR/MS tiene que ver primordialmente con la identificación y modificación de los supuestos básicos que van caracterizando la propia evolución de los modelos. Todos estos supuestos, sean evidentes o establecidos por la vía de la intuición, entran en funcionamiento de acuerdo al orden mental que cada individuo tenga predeterminado de ellos de acuerdo a su experiencia empírica.¹² Lo anterior identifica la configuración del modelo conceptual que cada administrador tiene implícitamente para ofrecer un planteamiento lógico y a la vez una solución para cada problema. Así pues, los modelos de OR/MS deben ser vistos como los mecanismos mediante los cuales se materializan las estructuras implícitas de las experiencias de las personas.

En la Figura 1.8 se resume el proceso de modelación matemática expuesto en este apartado proveniente de la metodología del Dr. Laurence Moore, profesor de la Universidad Estatal de Virginia.

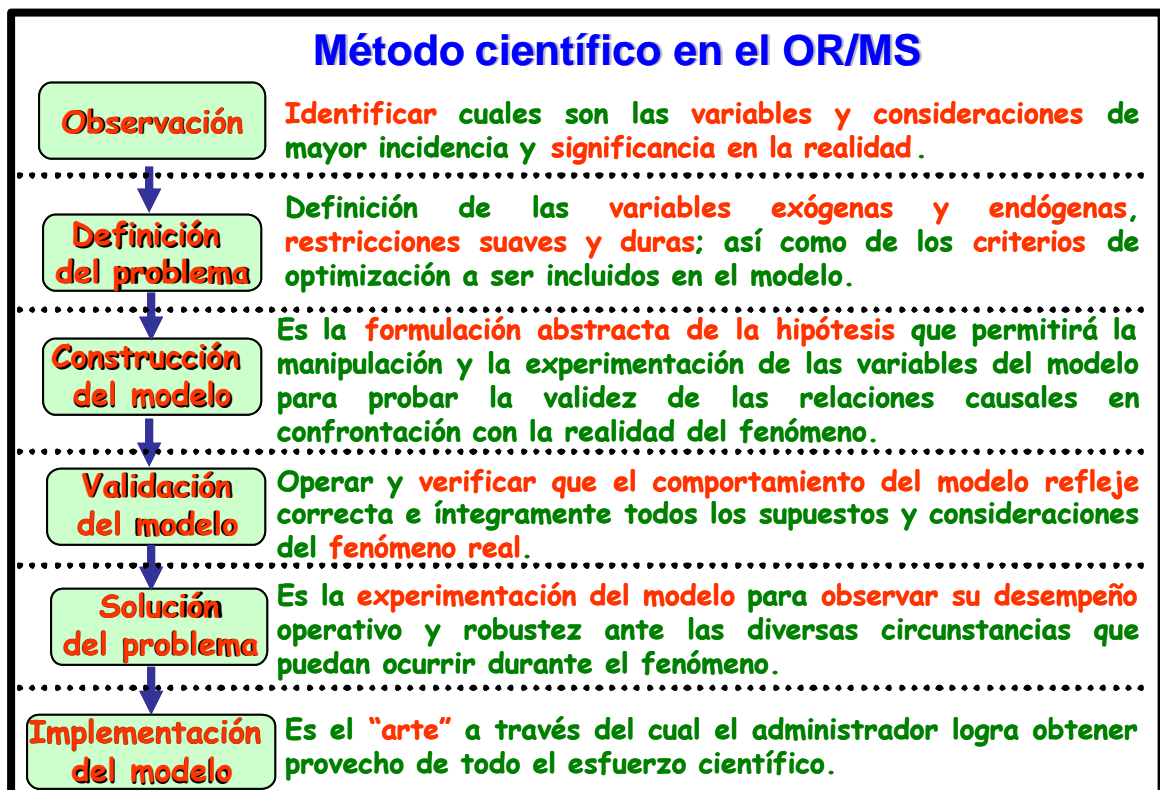


Figura 1.8 Metodología aplicada para el ciclo de modelación.¹³

¹² Berry, William. (1990), Management Decision Science, 1thEd, Irwin, EUA, pag 543.

¹³ Moore, Lawrence. (1993), Management Science, Allyn and Bacon, EUA, pag 4.

1.5 Definición de la función logística.

La “logística” se refiere a todas las funciones de una industria que puedan estar situadas hacia el interior de un negocio o bien subcontratadas hacia el exterior, a través del uso de proveedores de materiales o servicios contribuyendo todos ellos al flujo de información y productos que están siendo requeridos por los clientes. En este sentido, de nada sirve establecer un plan logístico estratégico si no se dispone de una logística operativa eficiente. Se requiere ligar ambos de igual modo que ocurre cuando se equilibran los controles estratégicos de una empresa que se ocupan del largo plazo, con los controles financieros que miden el desempeño del negocio en el corto plazo.

La principal búsqueda de la logística es el de asegurar y mantener una operación particular mediante la cual se cumplan eficientemente los objetivos definidos por el negocio. A la logística podemos definirla como un sistema influenciado por las restricciones y tendencias del medio ambiente económico, político, social y ético que busca asegurar tener el producto correcto, en el momento correcto, en el lugar correcto, en la condición correcta, en la cantidad adecuada y al mínimo costo posible, a través de la utilización de recursos humanos, tecnológicos y financieros, para asegurar una óptima relación costo y servicio, contribuyendo a la rentabilidad de la empresa a través del tiempo.

Es indudable que la práctica de la logística es aún inexacta. No obstante, la aplicación del OR/MS contribuye a optimizar la práctica de la misma. En la aplicación del OR/MS en la logística, el punto de partida es la modelación ya que esto presupone la identificación de los datos relevantes así como la comprensión de los problemas a ser resueltos.

Desafortunadamente, en situaciones reales, el administrador logístico no siempre dispone de toda la información necesaria para modelar un problema. Por tanto, es necesario tener la sensibilidad y la rigurosa comprensión del problema como para poder determinar cuándo un dato resulta relevante y cuándo no.¹⁴ En las áreas de logística en la actualidad, existe un incremento en la complejidad de los problemas, además que estos tienden a ser cada vez más diferentes entre sí. Lo anterior dificulta o imposibilita la aplicación complementaria de la experiencia pasada.

¹⁴ Berry, William. (1990), Management Decision Science, 1thEd, Irwin, EUA, pag VII.

Los administradores encargados de la función logística que no dispongan del conocimiento relacionado al OR/MS, estarán expuestos a su suerte o únicamente a su experiencia. Con la aplicación del OR/MS, la función logística tiene una mejor oportunidad para encontrar una solución integral y eficaz a los problemas comunes que se enfrentan comúnmente en el área. Los nuevos retos de la logística, deben ser enfrentados mediante la activación del proceso pensante que permita el desarrollo de soluciones óptimas y efectivas cuantitativamente hablando; pero sobre todo, también satisfactorias en el contexto social y humano que es en donde se van a implementar. La optimización a la cual nos referimos en el contexto de la logística tiene que ver en un sentido general, con la asignación y utilización eficiente de los recursos limitados para lograr objetivos concretos.

El análisis cuantitativo desarrollado para la optimización de los procesos logísticos de un negocio por sí mismos deben ser vistos de manera trans-funcional. Esto quiere decir que al igual que sucede con el fenómeno de óptimos locales y de óptimos globales al más puro estilo del álgebra polinomial; de igual modo puede ocurrir una condición de operación óptima de manera aislada para una área que podría significar una pérdida de efectividad y eficiencia (sub-optimización) referido al entorno o la visión global del negocio. La logística es en esencia un proceso trans-funcional e interdisciplinario. La trans-funcionalidad ocurre en el momento en que los objetivos que se buscan alcanzar en los procesos de logística no pueden ser vistos de manera independiente en cada área del negocio, sino que buscan ser comunes y congruentes al objetivo global de la organización.

Así pues, para que el OR/MS pueda ser aplicado en la logística se debe partir de la existencia de un problema que resolver. Luego, para que un problema realmente exista, se requiere antes que nada, que exista un individuo que lo reclame como tal. Además, deben de existir al menos un par de alternativas de solución al problema con repercusiones distintas; de otra manera no hay problema que resolver. Finalmente, el número de alternativas para que un problema pueda ser resuelto debe ser finito.

1.6 Aplicación del OR/MS en la práctica de la logística.

En el tema de “Modelación Matemática” ya antes expuesto, se estableció que una buena estrategia para iniciar el desarrollo de un modelo de OR/MS es a partir de la alineación de este último respecto al modelo conceptual que el administrador tenga de la realidad. Así pues, el modelo de OR/MS primero debe cubrir y reproducir el modelo conceptual que el administrador tiene de la realidad y luego entonces, como parte de su proceso evolutivo, empezar a enriquecer el modelo para resolver aquellas imperfecciones u omisiones que el administrador tenga en su modelo conceptual que sirvió como punto de partida.

No es justificable forzar el inicio con modelos de alta complejidad en aras de cubrir una fidelidad que simplemente nunca se ha tenido en el pasado. En su lugar resulta más conveniente premiar la robustez de un modelo, lo cual permitirá que éste sea sensitivo a los cambios o modificaciones que pudieran irse generando en los supuestos que lo caracterizan. La robustez será el cimiento que permitirá que el modelo se prepare para su proceso de enriquecimiento evolutivo.

Incluir en el modelo todos aquellos juicios subjetivos que los administradores utilizan en sus modelos conceptuales, otorga un sentido de pertenencia al modelo de OR/MS en la mente del administrador. Ciertamente esto puede reflejar una cierta subjetividad pero le ofrecerá al modelo una serie de atributos para el manejo de aspectos cualitativos logrando que el comportamiento del modelo sea predecible de acuerdo a la guía útil que le ofrezca el administrador dada su experiencia. La única advertencia a este respecto será que esa misma guía nunca deberá permitir acotar el modelo impidiéndole la búsqueda de soluciones inexploradas o desconocidas para el administrador.

Hablar de la viabilidad del OR/MS para su aplicación en la logística, consiste en determinar cuándo un modelo es lo suficientemente viable como para considerarlo útil en la práctica. Dicho de otro modo, una solución matemáticamente óptima, no equivale a que sea necesariamente viable para implementar en la práctica de la logística. En la práctica de la logística es frecuente que ocurra el rechazo del modelo por parte del administrador. Para asegurar que un proyecto tenga éxito necesitamos provocar que el modelo matemático sea para el administrador, una extensión real de su habilidad, concepción y experiencia para resolver los problemas.¹⁵

¹⁵ Berry, William. (1990), Management Decision Science, 1thEd, Irwin, EJA, pag 12.

De lo anterior podemos advertir que algunas de las características que los modelos deben incluir son:

1. Pertinencia y aplicabilidad respecto al problema que se busca resolver.
2. Claridad con la cual logra representar la abstracción de una realidad.
3. Adecuada parametrización.

De la parametrización podemos agregar que ésta se refiere a la interfase que le permitirá al administrador la oportunidad de hacer analogías entre el modelo matemático y su modelo conceptual de la realidad, lo cual redundará en un sentido de pertenencia y pertinencia.

No obstante todo lo anterior, en la práctica de la logística, el proceso de abstracción y modelación no es suficiente. Es necesario mantener el sentido de la realidad en virtud de que lo que finalmente se pretende, es utilizar un lenguaje abstracto que permita la comprensión y optimización de dicha realidad logística. Dado lo anterior, la práctica de la logística no está únicamente sujeta al análisis absolutamente objetivo y rigurosamente exacto de los modelos de OR/MS. La logística se distingue en que tiende a ejercer un contacto auténtico con la realidad siendo que su objetivo no es científico sino más bien pragmático. Está orientada hacia finalidades prácticas, debiendo tener en cuenta las necesidades y exigencias sociales.

En la práctica común de la logística, los administradores toman decisiones no en función a criterios estrictamente matemáticos de optimización, sino más bien buscando encontrar una solución que mejor interactúe y converja con las decisiones que otros administradores estén por igual llevando a cabo en el mismo tiempo y espacio. Así pues, los administradores toman decisiones en función a criterios de política organizacional más que de optimización matemática. La racionalidad de lo anterior es que finalmente, el conocimiento de cualquier individuo es solo cuando mucho una versión individual y simplificada de la realidad.

Los asuntos de política organizacional presentes en la problemática logística, no significan que debamos dejarnos guiar solo por la intuición individual o colectiva y renunciar a toda posibilidad de raciocinio que permita trascender a los paradigmas organizacionales. Aún en esta situación, podemos recurrir a afirmar que los modelos efectivos son aquellos que sin dejar de ser formales, incluyen el conjunto de conocimientos de varios individuos. La práctica de la logística viene desarrollando actividades y funciones que quizás en el pasado resultaron suficientes. No obstante en el contexto de negocios de altos volúmenes y bajos márgenes, se hace indispensable el desarrollo e implementación de nuevas formas de organización que permitan aumentar la productividad de los recursos de las empresas.

1.7 Comentarios finales: El arte en la aplicación del OR/MS y el arte en la práctica de la logística.

Debemos aclarar que la ciencia y el arte no son excluyentes, sino complementarios ya que es precisamente el arte lo que motiva y prolonga el estudio de la ciencia ante la mirada evaluativa de su propio rendimiento. No obstante lo científico y matemático que pueda resultar un modelo abstracto, en frecuentes ocasiones se llega a ellos a través del proceso de la intuición. De hecho este proceso de intuición aparece prácticamente en muchos otros campos de la actividad creativa humana. Esta aproximación hacia la solución de los problemas contrasta significativamente con la naturaleza explícita que tiene un modelo matemático el cual representa el resultado de este primer paso intuitivo.

Así pues, el arte en la aplicación del OR/MS tiene que ver con el desempeño y el rendimiento en la aplicación del conocimiento. El arte existe a través del uso de la intuición como respuesta racional del porqué debemos empezar con la definición matemática de aquellos supuestos más simples los cuales serán cruciales y altamente sugestivos para el desarrollo más adelante de aquellos otros supuestos que no lo sean tanto en un inicio. El aporte real del OR/MS existe, no como la ciencia de la optimización ni de la simulación, sino más bien como aquel arte volviéndose ciencia preocupada por el razonamiento lógico y comprensión rigurosa de los problemas de la administración en lo general y de la logística en lo particular a través del uso de modelos formales.

Por otro lado, aunque la logística inevitablemente incluye como parte de su estructura de conocimiento al OR/MS, no puede desligarse de todo aquello que sustente la aplicación práctica de la misma. El OR/MS se rige por la racionalidad, pero en el ejercicio de la logística se deben considerar aspectos subjetivos que matizan los diversos criterios o reglas de decisión que irrevocablemente desembocan en la necesidad de tener que elegir una solución a los problemas. El arte en la práctica de la logística, se materializa en el ejercicio de la toma de decisiones. Aunque parezca poco ordenado el contexto anterior en el cual se manifiesta un cierto grado de inexactitud, esto no debe impedir que la administración logística pueda acceder a la interpretación científica si consigue rebasar el uso de la intuición y del empirismo, para llegar progresivamente a lo racional y entonces considerarla como una ciencia aplicada a través de la acción humana.¹⁶

La logística está esencialmente orientada hacia la acción y no se limita a acumular conocimientos, sino que trata de emprender y transformar las situaciones buscando alcanzar el logro de los objetivos del negocio; y

¹⁶ Rodríguez, Joaquín. (1998), *Introd. a la Administración con Enfoque de Sistemas*, 3era Ed, ECAFSA, México, pag 254.

es precisamente aquí en donde interviene el arte en la logística. Así pues, el arte equivale a la intuición que debe poseer el administrador que le permita seleccionar aquella alternativa más viablemente implementable en la práctica, de entre todo el conjunto de soluciones que resultan ser óptimas en lo racional y en lo cuantitativo.¹⁷ Dicho de otro modo, consiste en elegir la solución óptima desde una perspectiva global de entre las soluciones óptimas locales. Como puede verificarse, aún en esta circunstancia en la que se habla de arte, aquí también está presente la racionalidad.

Así pues, en la práctica de la función logística existe una complementariedad entre ciencia y arte debido a que los conocimientos del administrador, deben prolongarse a través de la destreza y la capacidad propia del ejercicio de su profesión. Es decir, llega el momento en que la objetividad y la racionalidad que ofrece el OR/MS finaliza su encomienda y el administrador legítimamente tendrá que hacer despliegue de su intuición y experiencia para el mejor aprovechamiento de los recursos científicos y racionales.

En el desarrollo de los modelos de OR/MS, se debe ser objetivo evitando que la implementación computacional llegue a sustituir la problemática real. Sería lícito por ejemplo, dejar temporalmente fuera de un planteamiento matemático ciertos aspectos con la finalidad de partir de un modelo sencillo para evolucionarlo eventualmente. Pero, no sería lícito hacerlo solo por supuestos de eficiencia y "performance" computacional en detrimento de su viabilidad y rendimiento en su aplicación práctica.¹⁸

A continuación se muestra en la Figura 1.9 un diagrama esquemático de los conceptos que han sido abordados en este primer capítulo que estamos ya concluyendo, esto con la finalidad de contextualizar la relación del área problemática que es la logística con el área de aplicación que es el OR/MS. En dicha figura puede ser entonces establecida la relación entre la logística y el OR/MS. En el siguiente capítulo 2, abordaremos la problemática que guarda el factor educativo como fuente de desarrollo para las competencias y habilidades requeridas en el individuo que hacen posible la aplicación práctica del OR/MS para solucionar los problemas de la logística en las empresas. En lo referente a la exposición de los problemas típicamente encontrados en las áreas de logística, estos se cubrirán luego en el capítulo 3 en donde se revisarán sus diferencias y similitudes y a la vez se establecerá el problema en el cual nos concentraremos para el desarrollo y tratamiento de la tesis doctoral.

¹⁷ Render, Barry. (2000), Quant itative Analysis for Management, 7thEd, Prentice Hall, EUA, pag 15.

¹⁸ Berry, William. (1990), Management Decision Science, 1thEd, Irwin, EUA, pag 535.



Figura 1.9 Diagrama esquemático acerca de la relación entre el OR/MS y su aplicación en la logística

El poseer una mente vigorosa no es suficiente; lo importante es no tardar en aplicarla.

Rene Descartes

2. El factor educativo en la aplicación de la investigación de operaciones.

2.1 Introducción: el proceso de enseñanza y aprendizaje en el OR/MS.

Podemos definir que "enseñar" es toda aquella acción intencional tendiente a favorecer el aprendizaje. El problema del aprendizaje equivale al problema de cómo acceder al conocimiento. El aprendizaje requiere no solo de un método sino también del manejo de las experiencias y recursos previamente aprendidos. Así pues, para seguir aprendiendo se requiere comprender lo antes "aprendido".

Un aprendizaje realmente "significativo" es aquel que rompe el equilibrio o el "status quo" psíquico del individuo, vinculando el conocimiento nuevo con el previo mediante una relación no memorizada, sino construida. El real aprendizaje, contribuye a crear nuevos significados, es transferible a nuevas situaciones para solucionar nuevos problemas y al mismo tiempo actúa como agente motivador para emprender aprendizajes nuevos de manera autónoma. Es en ese sentido de autonomía mediante el cual se motivan las actitudes de investigación. Es indudable que obtener resultados por investigación propia es más valioso, que estudiar lo que otros han descubierto.

La existencia y generación de la información, conocimientos y experiencias es lo que estimula el proceso comunicativo entre las personas. Así pues, sucede que el medio detonante de toda enseñanza inicia a partir del proceso llamado "comunicación". El proceso de comunicación prueba que el conocimiento, no es tan solo una copia de la realidad solitaria de una persona, sino que se construye por el interactuar del ser humano con los demás. Por tal motivo, el aprendizaje es un proceso necesariamente participativo, es decir, el que aprende interviene y contribuye en su propia educación.¹⁹ Así pues, el estudiante participa activa e inevitablemente en la producción de su propio aprendizaje.

Con lo mencionado ya antes, resulta indudable que el elemento sustentable y progresivo del aprendizaje tiene que ver con el desarrollo de la habilidad cognoscitiva para el manejo del lenguaje que permita comunicar las ideas a los demás. La enseñanza del OR/MS tiene como objetivo un aprendizaje encaminado hacia el desarrollo de "competencias" específicas las cuales serán objeto de exposición

¹⁹ Tenti, Emilio. (2000), Educación Participativa, Argentina, disponible en <http://www.utdt.edu/eduforum/>.

más adelante en el presente capítulo. Estas “competencias” antes referidas, no equivalen a reemplazar una idea por otra, ni tampoco a acumular conocimiento. Se trata de una acción encaminada a promover la transformación de un pensamiento activo y crítico que permita describir y modelar la realidad.

En el caso particular del OR/MS, existe una dificultad en su aprendizaje debido esencialmente a la ausencia de una comprensión significativa, es decir, se tiene a dos personas hablando del mismo objeto pero con perspectivas diferentes. Este elemento de subjetividad personal nos conduce a afirmar que el aprendizaje en el individuo se construye a partir de su desarrollo psíquico y social histórico previo. Nuevas formas de conocimiento se vinculan y relacionan a partir de las anteriores. Los conocimientos para modelar no se transmiten mecánicamente, sino representan una mezcla de conocimientos previos y de la experiencia práctica. Aunque no podemos suponer que para el alumno le sea suficiente su satisfacción intelectual, debemos afirmar que el aprendizaje del OR/MS es algo que hace el estudiante y no algo que se le hace a él.

El aprendizaje en el OR/MS, no equivale a comprender las palabras. A lo que nos referimos por comprensión tiene que ver con las ideas. La relación entre pensamiento y palabra es un proceso, un continuo ir y venir del pensamiento a la palabra y de la palabra al pensamiento. Así pues, las palabras cumplen cabalmente su cometido al servir como instrumento para comunicar las ideas. Por lo tanto desde el punto de vista formal, debemos entender por comprensión toda aquella suma de características que hacen único y diferenciable a un objeto. Este significado de comprensión del objeto(s) es aplicable tanto en los modelos conceptuales como también en los matemáticos.

El aprendizaje no es la educación de los saberes que enseña “cosas” sino una educación que forme “competencias” en el individuo.²⁰ El concepto de “competencia” al que se refiere Aguerro tiene la misma connotación al que nos referiremos en el apartado 2.3 cuando se expongan dichas “competencias”. El aprendizaje del OR/MS debe estar orientado a promover el uso del razonamiento. Es aquella educación que se centra en el desarrollo de habilidades interdisciplinarias que forme personas competentes que logren ir más allá de los conocimientos y trasciendan en el ámbito de la actuación. Esto último servirá como un medio activo para la solución de los problemas de las empresas ante la necesidad de tener que competir en un mundo global y en el cual resulta crucial la optimización de las operaciones. Precisamente es en la actuación y en el desempeño del individuo en donde se despliegan las competencias que buscamos.

²⁰ Aguerro, Inés. (1998), La Educación del tercer milenio, “Revista Academia”, Argentina, pag 3-5.

2.2 La práctica profesional del OR/MS en las empresas.

La Universidad ha tenido la misión de proveer a los individuos de los conocimientos necesarios para acceder al conocimiento de la época. Aunque el aprendizaje del OR/MS se gesta en las Universidades, este culmina una vez que el individuo busca llevar lo aprendido en la academia a su aplicación en el plano práctico profesional de las empresas. Este proceso de aprendizaje continúa inacabablemente en el interior de las empresas tanto de una manera individual como también de manera colectiva y colaborativa. El ambiente colaborativo resulta un ingrediente esencial en la práctica del OR/MS.

En términos generales, podemos decir que las organizaciones empresariales diseñan sus acciones sobre la base de sus habilidades para adecuadamente aplicar sus recursos y así satisfacer sus requerimientos internos y la demanda de sus clientes externos. En la medida en la cual estas habilidades se alejen de la condición de optimalidad, en esa misma medida las empresas habrán de tener que costear los incrementos de capacidad que sean suficientes para compensar esa falta de habilidad antes mencionada.

Este incremento de capacidad en las empresas antes mencionado, afecta al menos la rentabilidad de las empresas. No obstante lo anterior, esto compromete un capital innecesario que bien pudiera ser utilizado para poner en marcha estrategias de mayor alcance y así poder aspirar a acceder a alguna oportunidad o ventaja competitiva en el futuro. Las condiciones actuales han cambiado y la globalización impide ahora que el costo de estas ineficiencias puedan ser aplicadas al precio de los productos que los consumidores demandan.

Optimistamente al suponer que los profesionistas dispongan de las competencias para la práctica del OR/MS, luego aparece otro obstáculo. En la actualidad se presenta una subcalificación de los puestos de trabajo ocasionando que “un porcentaje alto de los profesionistas se desempeñen realizando tareas de nivel operativo”.²¹ La opinión del postulante es que en la práctica profesional del OR/MS en las empresas existe una subutilización de los profesionistas lo cual propicia un desperdicio de posibilidades. Esto último no es exclusivo de la profesión del OR/MS, sino en general de las especialidades relacionadas con el desarrollo y la aplicación de la tecnología. Dicho de otro modo, no importa cuanta teoría se enseñe y aprenda en las universidades, mientras no sea demandada y aplicada en las organizaciones, ésta no tendrá ningún valor.

²¹ Gomez, M. (1998), Inserción ocupacional de los egresados universitarios, “IV Congreso Nacional de Estudios del Trabajo”, Argentina, pag 9.

Lo anterior no significa que el origen del problema esté en las empresas. Ciertamente las empresas podrán decir que este aprendizaje se gesta en las universidades; dirán que es el lugar en el cual se aprenden las “competencias” y el conocimiento que más adelante, en el ámbito empresarial, se transformará en materia de soluciones a los problemas. No es extraño revisar en el caso mexicano, que las empresas hayan podido hasta ahora sobrevivir sin la necesidad de aplicar el OR/MS. Quizás, no ha habido en el pasado una necesidad para hacerlo, quizás todo se reduce a que le ha faltado a la nación estar expuesta a una economía de guerra la cual obligue a tener que optimizar las operaciones para lograr hacer mas con menos.

De esta manera, estamos encontrando que el punto de partida para el problema se sitúa ciertamente en el aprendizaje, pero más específicamente en aquel que puede llegar a influir de una manera general en la cultura de una nación y no solamente en una empresa. Estamos hablando de un aprendizaje que logre desarrollar “competencias” que evite la dependencia tecnológica, industrial, económica y porque no decirlo también, política hacia los socios extranjeros.

2.3 Las competencias requeridas para la aplicación del OR/MS.

La habilidad o competencia buscada en el OR/MS es la aplicación de conocimientos para la solución de problemas específicos a la administración de las empresas. En base a la empírica docente del postulante con 15 años de experiencia en escuelas de postgrado en la ciudad de Monterrey, a continuación se exponen “tres competencias” que se consideran básicas para la práctica del OR/MS:

- Competencia de Abstracción: abstracción y razonamiento inductivo – deductivo.
- Competencia Intuitiva: sensibilidad en su aplicación práctica.
- Competencia Comunicativa: saber extraer y comunicar las ideas.

Estas “tres competencias” no funcionan por separado, sino que se integran de una manera coherente logrando de esta manera la viabilidad en la práctica. Las habilidades de abstracción y de intuición proyectan las de implementación; las de intuición y comunicación proyectan las de actuación y las de abstracción y comunicación proyectan las de proposición. Esto se muestra en la Figura 2.1.

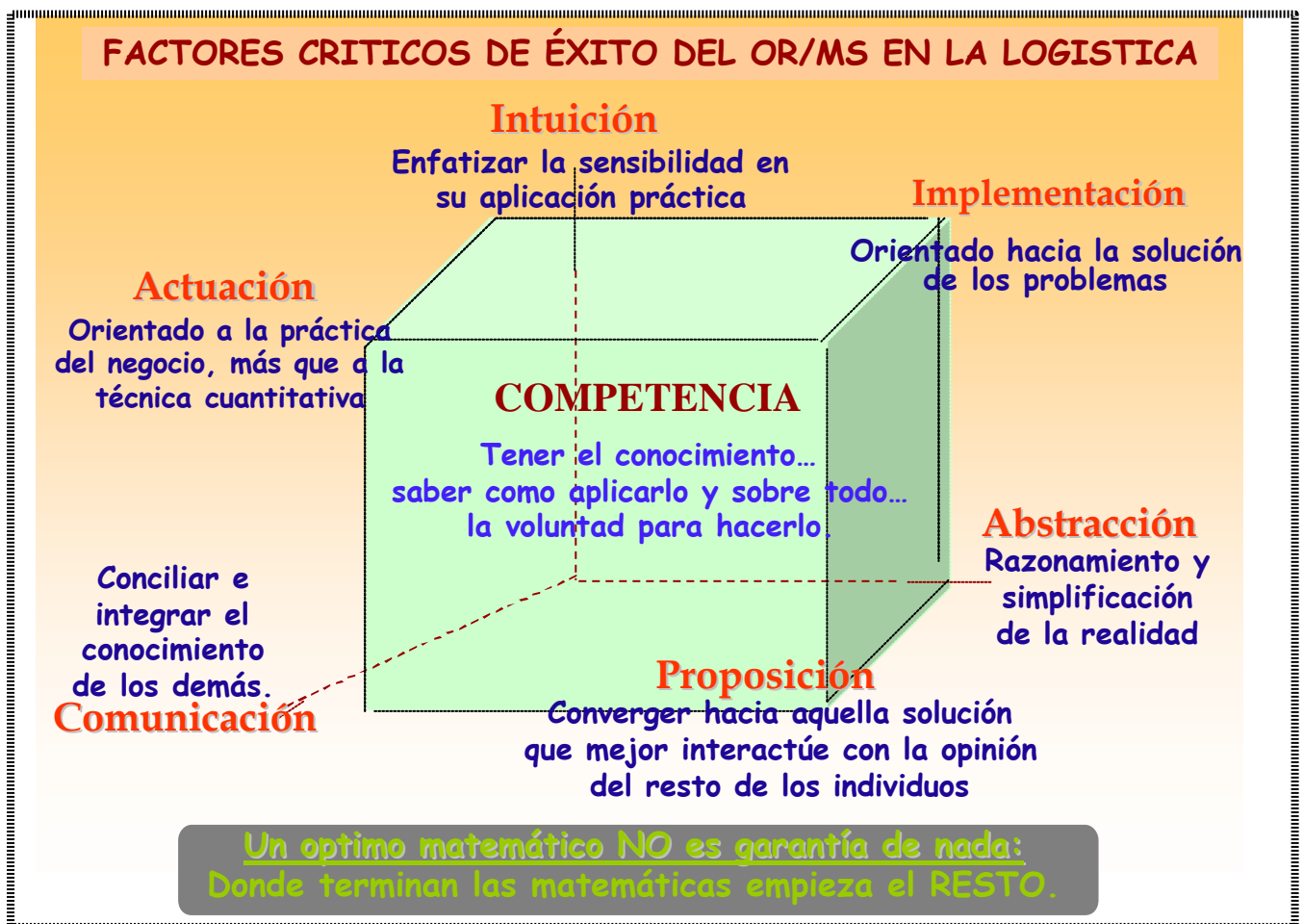


Figura 2.1 Competencias buscadas en el OR/MS.

Para lograr desarrollar estas competencias se requieren esencialmente tres ingredientes:

1. El docente: se requiere por parte de él promover un aprendizaje apoyado en el lenguaje, la participación, la elaboración de alternativas, la discusión y la toma de decisiones. Evaluar formativamente el desarrollo de competencias y no la memorización de palabras. Establecer una correlación entre el lenguaje oral y el lenguaje simbólico-abstracto, propiciando el entendimiento de la lógica y de las matemáticas. Facilitar la conexión de diferentes saberes, darle significado al aprendizaje y fomentar la capacidad para identificar y solucionar problemas con actitud creativa. Finalmente, deberá educar no solo el intelecto sino también la voluntad; ¿De que sirve un conocimiento si no se le encuentra un provecho significativo ?

2. El alumno: se requiere su participación real, debe aprender a pensar, modificar sus hábitos de pensamiento, actitud y disposición hacia el cuestionamiento, la investigación y la construcción del conocimiento. Debe asimilar una cultura de innovación, trabajo en equipo, inclinación a tomar riesgos, experimentar y evaluar consecuencias. Debe participar activamente en el desarrollo de su propia capacidad y destreza para futuros aprendizajes, aprendiendo a descubrir por sí mismo las cosas y a ser creativo en la solución a problemas. Así pues, la educación hacia el alumno se desarrolla en dos direcciones que se combinan armónicamente, su intelecto y su voluntad.²² Todo lo anterior implica transformar el aprendizaje en conocimiento y el conocimiento en competencia a través de la actuación y de la voluntad.

3. La tecnología: todas las áreas de la actividad humana se han transformado gracias a la tecnología, no obstante, el método de enseñanza de las matemáticas en las escuelas ha metabolizado a las tecnologías sin resultar estas en modificaciones sustanciales. La primera reacción que algunos profesores experimentan frente a la tecnología, es solo para utilizarla para continuar viejos métodos de enseñanza. Es decir, se pierde la oportunidad que brinda la tecnología para evolucionar el contenido de lo que se enseña en la academia y así descubrir nuevas posibilidades y capacidades de desarrollo en el aprendizaje.

²² Mota, Flavio. (1998), El deber ser y el querer ser en la Educación, "Revista Academia", México, disponible en <http://kepler.uag.mx/temasedu/deberser.htm>

La generalidad de los maestros se resisten al uso de la tecnología ya que ello les requiere tener que estar preparados para tratar con más alternativas, opciones y preguntas. En un estudio reciente que se desarrolló a lo largo de seis años en West Virginia, un tercio de la mejora en comprensión de lectura y razonamiento de las matemáticas fue atribuida al uso de nuevas tecnologías.²³

Desde una perspectiva mercantilista, el mecanismo de intervención de un alumno que busca un aprendizaje ocurre a través del proceso de compra. Es decir, ellos participan en el servicio comprando o dejando de comprar y así instrumentan el envío de señales del mercado al sector. No obstante, desde la perspectiva del postulante, los alumnos no son consumidores o clientes, sino más bien son coproductores activos del servicio con sus propios derechos y obligaciones. Así entonces en el modo de ver mercantilista, la educación estaría operando de una manera que no sería congruente con el deber ser orientado hacia una participación más activa por parte del alumno.

²³ Dale, Mann. (1999), Achievement gains from a Statewide Comprehensive Instructional Program, Milken Family Foundation, EUA, pag 14.

2.4 El desarrollo de competencias en México.

Hablar de tener que transformar la cultura en México nos remite directamente al contexto de la educación. No hay duda que la calidad de la educación propicia mejores mecanismos de aprendizaje. Hablar de calidad en la educación no equivale a pensar en niveles de escolaridad. Estamos hablando de un cambio en el comportamiento actitudinal de las personas en lo individual y de un cambio en su orientación colaborativa en lo grupal. En ambos casos se requerirá estar capacitado para hacer frente a situaciones problemáticas en las cuales se requiere la puesta en marcha de soluciones de fondo.

La sociedad en México, ha insistido en la demanda de un conocimiento pragmático. Esta postura pareciera que más que “competencias”, lo que la sociedad demanda es tan sólo la posibilidad de un mejoramiento en el nivel de vida de la población. Lo anterior deja muy lejos la posibilidad de un cambio de cultura. No podemos afirmar que actualmente la sociedad muestre un positivismo hacia la formación de una cultura competente en el entorno global, sino desafortunadamente quizás sólo una actitud de acomodamiento y conformismo respecto a lo que se está gestando en el mundo de acuerdo a las nuevas reglas económicas y políticas con las cuales se tiene que coexistir.

En términos generales y apeándonos a las evidencias disponibles y conocidas, las políticas educativas en México han hecho énfasis solo en aspectos de cobertura ya que se registra una de las cifras más altas en América Latina en cuanto al porcentaje de la población que atiende el sistema educativo.²⁴ No obstante en el asunto de la calidad educativa los datos académicos apuntan a considerar a México como un país de reprobados.²⁵

En el mes de octubre del año 2001, los editoriales de varios periódicos del país han hablado a este mismo respecto. Salen a la luz pública los resultados internacionales que en materia de calidad en la educación se refiere. En función a la muestra de estudiantes utilizada en el análisis comparativo internacional, resulta específicamente en el caso de los estudiantes mexicanos un bajo nivel de capacidad y de razonamiento abstracto en el área de matemáticas²⁶.

²⁴ Dávila, S. (1999), El papel del docente en la calidad educativa, México, en <http://www.nalejandria.com/akademeia/sdavila/>

²⁵ Marchand, Horacio. (2001), ¿ Una nación en Riesgo ?, EGADE ITESM, México.

²⁶ Delgado, Mónica. (2001), SEP difunde resultados de la evaluación Internacional: Entrevista al Secretario de Educación Reyes Taméz, "Periódico el Norte", México, pag 3A.

El resultado actual solo es la consecuencia lógica de una política educativa que no tuvo previsión de las tendencias de globalidad y avance tecnológico dentro de la competencia económica actual. El país se halla sujeto y dependiente del exterior en aspectos que van desde el simple asesoramiento tecnológico hasta el condicionamiento económico y político.

El 16 de Octubre del 2001 se publicó en un diario de circulación nacional los resultados de la evaluación de México en matemáticas y ciencia desarrollados por parte de la Asociación Internacional para la Evaluación del Logro Educativo (IEA). El resultado es que México está en los últimos lugares. El editorialista pregunta entonces ¿Hay relación entre el mal desempeño educativo en matemáticas y el nivel de desarrollo de un país? Sin duda, responde el editorialista.²⁷

La relación específica que guarda el tema del desarrollo de “competencias” en la educación en México con respecto a la tesis doctoral se refiere a la necesidad de establecer antes que nada un supuesto básico que debe existir en toda sociedad que pretenda dar solución a problemas como el que será atendido en la presente tesis doctoral. Es decir, antes que un algoritmo o propuesta matemática, lo primero que debe existir para dar solución a un problema, es la presencia de un individuo (tomador de decisiones) que atienda el problema como tal, lo comprenda en términos de un modelo conceptual, relacional y lógico; logrando identificar la información que sea relevante. Cubierto el supuesto anterior, luego entonces ahora sí podremos pasar al tema de las matemáticas y de los algoritmos. Dicho de otro modo, el punto de partida está en el desarrollo del modelo y luego después en el algoritmo.

De la argumentación anterior, es fácilmente comprensible que el problema particular de la educación y las “competencias” antes descrito es por sí solo materia suficiente para un estudio por separado. Aunque por tal motivo, este tema particular escapa al alcance de investigación de la presente tesis doctoral, nos hemos permitido realizar un estudio de campo que pretende estudiar un poco más de cerca dicho fenómeno. A continuación en el siguiente apartado se explica y se expone dicha problemática.

²⁷ Quintana, Enrique. (2001), Matemáticas y Economía, “Editorial el Norte”, México, pag 5b.

2.5 Caso de estudio: La aplicación de las matemáticas por parte de los profesionistas de las empresas del área metropolitana de la ciudad de Monterrey.

En este apartado se abordará el tema referente a las razones por las que el OR/MS ha tenido dificultades para ser aplicado por los profesionistas. Asumiendo que el término OR/MS no es del dominio público, decidimos substituir “OR/MS” por “Matemáticas” sobre el entendido de que ambos conceptos son razonablemente equivalentes para efectos del estudio de campo. Así pues, el objetivo del experimento consiste en identificar aquellos factores que causan que las matemáticas no sean aplicadas por los profesionistas de las empresas del área metropolitana de la ciudad de Monterrey (AMM). El diseño experimental propuesto por el postulante para dar respuesta a esta interrogante es a través de un estudio de campo mediante un instrumento tipo cuestionario aplicado a dichos profesionistas.

Pedir la opinión al respecto de esta cuestión a los profesionistas, resulta desde la perspectiva del postulante, suficientemente pertinente; ya que son ellos mismos los que están en posibilidad de ejercer o no ejercer dicha habilidad.

Ahora bien, el diseño del cuestionario contiene un total de 64 reactivos que de aquí en adelante llamaremos variables. Estas 64 variables son el resultado de dos sesiones de lluvias de ideas que fueron llevadas a cabo a su vez con profesionistas de la escuela de la división de postgrado de la facultad de Contaduría Pública y Administración de la UANL durante el mes de Mayo del 2002. Cada una de estas variables representa una posible razón por la cual el OR/MS pudiera no estar siendo aplicado por los profesionistas en las empresas. A su vez, cada una de estas 64 variables aparece agrupada en un total de 8 constructos principales. Estos 8 constructos principales son los siguientes:

1. El alumno.
2. El maestro.
3. El método de enseñanza.
4. Las empresas.
5. El entorno económico.
6. La Cultura Nacional
7. Los antecedentes y las expectativas.
8. La naturaleza de las matemáticas.

Así pues, por un lado las 64 variables permitirán matizar detalladamente la opinión del profesionista respecto al problema tratado. Por el otro lado, los 8 constructos permitirán dar orden, estructura y dirección a los aspectos principales del problema. El instrumento de campo con las 64 variables contenidas en 8 constructos es aplicado para que los profesionistas registren en él su opinión del problema. En la aplicación del instrumento, cada una de las 64 variables deberá ser calificada de acuerdo al grado de importancia relativa que cada una de estas tenga para dar respuesta al problema en cuestión. La convención utilizada en la calificación de importancia fue utilizar el No. 1 para establecer la variable o razón principal y el No. 64 para aquella variable con menor importancia relativa.

De lo anterior, cada una de las 64 variables recibirá una calificación idealmente única que permitirá identificar similitudes y diferencias. Si sumamos los números que serán registrados para las 64 variables, indistintamente siempre resultará “2080”. Este último será nuestro código verificador para validar la medición. Con la finalidad de enumerar las 64 variables ya antes mencionadas, se muestra en la Figura 2.2 el prototipo del formato de encuesta que se aplicó para el estudio de campo.

Nombre	Empresa donde Trabaja			# de ocasiones en las que ha aplicado las Matemáticas para la toma de decisiones en los Negocios ?				
Escuela	Puesto			# de ocasiones en las que ha VISTO aplicar las Matemáticas para la toma de decisiones en los Negocios ?				
Carrera	Matemáticas en Carrera			# de ocasiones en las que ha VISTO aplicar las Matemáticas para la toma de decisiones en los Negocios ?				
Año	Matemáticas en Maestría			# de ocasiones en las que ha VISTO aplicar las Matemáticas para la toma de decisiones en los Negocios ?				
ENVIA HACIA QUIEN SOBRESALE	ENFOQUE HACIA LA IMPORTACION DE TECNOLOGIA	CULTURA TERCER-MUNDISTA	FALTA DE HABILIDAD PARA EL RAZONAMIENTO LOGICO DE LOS PROBLEMAS	FALTA DE TIEMPO (ALUMNO TRABAJA TIEMPO PARCIAL)	PROBLEMAS DE ALIMENTACION	LAS MATEMATICAS NO TIENEN APLICACION EN LA PRACTICA	ES MAS FACIL APLICAR EL SENTIDO COMUN QUE LAS MATEMATICAS	MI PUESTO NO REQUIERE TOMAR DECISIONES
FALTA DE TRABAJO EN EQUIPO		CORRUPCION Y CONFORMISMO	FALTA DE HABILIDAD PARA EL IDIOMA INGLES		FALTA DE INTERES Y VOCACION	LAS EMPRESAS NO REQUIEREN TANTA SOFISTICACION		NADIE APLICA LAS MATEMATICAS EN LA PRACTICA
MENOSPRECIO AL USO METODOS FORMALES	FALTA DE INTERES EN LA SOLUCION DE LOS PROBLEMAS DEL PAIS	FALTA DE CULTURA COMPETITIVA	FALTA DE MOTIVACION	CONFORMISMO	FALTA DE DISCIPLINA	LAS MATEMATICAS SON DIFICILES DE APRENDER	NO ME GUSTAN LAS MATEMATICAS	LAS EMPRESAS NO PREMIAN EL ANALISIS MATEMATICO EN LA TOMA DE DECISIONES
FALTA DE HABILIDAD PARA EL RAZONAMIENTO LOGICO DE LOS PROBLEMAS	FALTA DE ACTUALIZACION Y/O EXPERIENCIA	FALTA DE INTERES Y VOCACION	LA CULTURA NACIONAL	EL ALUMNO	LAS MATEMATICAS	NO CONOSCO A NADIE QUE APLIQUE LAS MATEMATICAS EN LOS NEGOCIOS	ES POCO PROBABLE QUE EN EL FUTURO REQUIERA APLICAR LAS MATEMATICAS EN LA TOMA DE DECISIONES	EN UN SENTIDO EFECTIVO EVITO UTILIZAR LAS MATEMATICAS Y PREFIERO GUIARME POR LA EXPERIENCIA
FALTA DE CULTURA COMPETITIVA		FALTA DE DISCIPLINA	EL MAESTRO		ANTECEDENTES Y ESPECTATIVAS	HE TENIDO MUY MALOS MAESTROS DE MATEMATICAS		MI PUESTO NO REQUIERE RESOLVER PROBLEMAS COMPLICADOS
FALTA DE TIEMPO POR EXCESO DE TRABAJO	MAESTROS MAL PREPARADOS PEDAGOGICA-MENTE	FALTA DE MOTIVACION	LA EMPRESA	EL METODO DE ENSEÑANZA	LOS RECURSOS ECONOMICOS	EN MI CARRERA NO APRENDI LA FORMA DE APLICAR LAS MATEMATICAS	QUISIERA APRENDER A APLICAR LAS MATEMATICAS EN LA TOMA DE DECISIONES	LA GENTE CON EXPERIENCIA PIENSA QUE LAS MATEMATICAS NO TIENEN UTILIDAD
FALTA DE CULTURA COMPETITIVA (MONOPOLIOS EMPRESARIALES)	CONFORMISMO EMPRESARIAL	ENFOQUE MAQUILADOR SIN POSIBILIDAD PARA LA INNOVACION	ENSEÑANZA MECANIZADA CON FORMULAS Y PROCEDIMIENTOS	PLANES DE ESTUDIO OBSOLETOS	FALTA DE PERTINENCIA EN LA SOLUCION DE PROBLEMAS REALES	FALTA DE MATERIAL ACADÉMICO	FALTA DE FOMENTO A LA INVESTIGACION	FALTA DE ESPACIOS ADECUADOS PARA EL APRENDIZAJE
FALTA DE TIEMPO (TOMA DE DECISIONES URGENTES)		FALTA DE INTERES PARA SU APLICACION	FALTA DE PROMOCION A DESARROLLO DE HABILIDADES COMPETITIVAS		NO SE CUBREN LOS PROGRAMAS DE ESTUDIO	FALTA DE TECNOLOGIA		FALTA DE VINCULACION ENTRE EMPRESAS Y UNIVERSIDADES
EXCESO DE TRABAJO	SE CUBREN PUESTOS CON GENTE NO PREPARADA	FALTA DE CONOCIMIENTO PARA SU APLICACION	EXCESO DE ALUMNADO	FALTA DE VINCULACION CON LAS NECESIDADES DE LAS EMPRESAS	UNIVERSIDADES CON BAJO NIVEL ACADÉMICO	FALTA DE MAESTROS BIEN PAGADOS	FALTA DE BECAS E INTERCAMBIO ACADÉMICO	SINDICALISMO, CORRUPCION Y DESVIO DE RECURSOS

Figura 2.2 Formato de encuesta para el estudio de campo para profesionistas de empresas del AMM.

A continuación, el diseño del experimento nos requiere normalizar los datos recabados en el instrumento de medición. Lo anterior, no es otra cosa que llevar la escala original de preferencia relativa de 64 niveles a una de solo 6 niveles. Esta nueva escala de 6 niveles se explica como sigue en la Tabla 2.1:

Tabla 2.1 Niveles de importancia relativa.

Escala de nivel	Significado de importancia relativa
0	En desacuerdo
1	Rara vez influye
2	Indiferente
3	A ser tomado en cuenta
4	Tiene mucha importancia
5	Es la causa raíz

La normalización de los datos recabados en el instrumento de campo, tiene la finalidad de concentrar el matiz de importancia relativa en un número menor de grados. Obviamente esta normalización resulta ser mucho más conveniente realizar una vez que el encuestado ha hecho el esfuerzo por establecer una diferenciación relativa de cada una de las 64 variables y no solamente sobre la base de 6 niveles de diferenciación.

Una razón más para llevar a cabo este proceso de normalización es para “corregir” los datos que estén mal registrados. Por ejemplo, ¿Qué sucedería cuando la medición de un cuestionario no sumará 2080 ?. En un caso como el anterior, es fácilmente entendible que el proceso de normalización ayudaría a eliminar el ruido que hubiera en la medición y así rescatar los datos. La formula utilizada para normalizar los datos se muestra a continuación:

$$G_n = \text{Int} \frac{64 \cdot G_e}{12}$$

Donde: G_n = Grado normalizado

G_e = Grado estandar

Ecuación para normalización de 64 a 6 niveles

Como puede apreciarse en la formula anterior, solo las variables que hayan sido seleccionadas con grado de importancia relativa “ G_e ” del 1 al 4 serán calificadas como “Causa Raíz”, es decir grado normalizado $G_n = 5$. Esto último es deliberado y resulta ser parte del diseño del instrumento.

2.5.a Identificación de la población de estudio y determinación del tamaño de la muestra.

La población de estudio está totalmente determinada en el experimento. Esta es el conjunto de profesionistas del área metropolitana de la ciudad de Monterrey (AMM) que se desempeñen actualmente laborando en alguna empresa de la ciudad y que tengan como actividad preponderante alguna función que requiera de toma de decisiones. A continuación se describen las características buscadas en los profesionistas:

1. Que hayan hecho sus estudios de licenciatura en Universidades del AMM después de 1980.
2. Que sus estudios académicos contengan asignaturas en las cuales tenga pertinencia la toma de decisiones de los negocios en general.
3. Que laboren en las empresas del AMM.
4. Que el puesto que desempeñen requiera de capacidad de análisis referido al proceso de toma de decisiones en los negocios.

La razón de la selección anterior, obedece a que aquí tenemos contenidos profesionistas que:

1. Recibieron su educación antes y también después del parte-aguas del OR/MS en EUA (1991).
2. Estos desempeñan responsabilidades suficientemente pertinentes como para llegar a requerir capacidades de análisis cuantitativo para la toma de decisiones.

Para el cálculo del tamaño de la muestra, partiremos naturalmente del tamaño de la población. Según el último censo de población del INEGI ²⁸, tenemos que el total de la población económicamente activa del estado de Nuevo León cuya actividad preponderante es desempeñarse como profesionista resulta ser un total de 67,380 individuos. Obviamente esta estimación de la población para nuestros propósitos es bastante razonable, ya que sabemos de antemano que no todos ellos tienen pertinencia con respecto a la toma de decisiones en la administración de los negocios. Así pues, el error experimental está cubierto y además protegido.

La probabilidad de suceso “p”, que se utilizó para el cálculo de la muestra fue del 5%. Este número proviene de la empírica del postulante, el cual más adelante en los resultados del experimento se sustentará. Este porcentaje preliminar equivale a observar la cantidad de alumnos que en las clases de

²⁸ Encuesta Nacional de Empleo. (2001), Base de Datos y Tabulados de la Muestra Censal, “XII Censo General de Población y Vivienda”, México, pag 45.

modelos cuantitativos a nivel maestría, afirman tener experiencia práctica en la aplicación de las matemáticas en la toma de decisiones en los negocios. La experiencia de 10 años del postulante, le permiten identificar un alumno con tales características por cada grupo de clase que normalmente oscila entre 15 y 20 alumnos en total. Más adelante este valor “p” podrá ser verificable en los resultados que arroje la investigación de campo.

Utilizando un margen de error del 5% y un porcentaje de confiabilidad al 95%, resulta un tamaño de muestra equivalente a 73 unidades experimentales. A continuación se muestra el desarrollo del cálculo.

N ? *Tamaño de la población* ? 67,380
 p ? *Probabilidad muestral* ? x/n ? 5% ? q ? $1-p$
 e ? *Error estandar permitido* ? 5%
 Z ? *Nivel de confianza* ? 95% ? z ? 1.96
 n ? *Tamaño de la muestra*

$$n ? \frac{N * p * q}{\frac{N * e^2}{z^2} ? p * q} ? 73$$

2.5.b Estadística descriptiva de la muestra.

A partir de aquí, se inició el estudio de campo durante los meses de Junio y Julio de año 2002. Fueron un total de 405 cuestionarios los que fueron aplicados por un grupo de colaboradores. A continuación se agregan una serie de tablas que servirá para conocer el origen de los datos.

Tabla 2.2 Clasificación de la muestra de acuerdo al tipo de empresa donde labora el profesionista.

# de Casos	
TIPO EMPRESA	Total
BANCA-SERV.FINANCIEROS	52
METAL-MECÁNICO	35
UNIVERSIDADES	27
AUTOSERVICIOS	24
IND. QUIMICA	23
CONSULTORIA	22
ACERO-CEMENTO-VIDRIO	21
TELEFONIA	18
AUTOTRANSPORTE	16
SERV.INTERNET	15
ALIMENTOS Y BEBIDAS	14
OTRAS	138
Total general	405

Tabla 2.3 Clasificación de acuerdo a la escuela de procedencia.

# de Casos	
Nom.Escuela	Total
ITESM	114
UANL FACPYA	108
UANL FIME	64
UDEM	44
UANL FCFM	27
UANL FCQ	21
OTROS	20
UR	7
Total general	405

Tabla 2.4 Clasificación de acuerdo a la carrera de procedencia.

# de Casos	
Nom.Carrera	Total
CONTADURIA	117
SISTEMAS	92
INGENIERIA	63
ADMINISTRACION	60
ING.INDUSTRIAL	34
OTRAS	17
MBA	12
MERCADOTECNIA	10
Total general	405

Tabla 2.5 Clasificación de acuerdo al año de graduación de la carrera profesional.

# de Casos	
Año	Total
1970-1974	2
1975-1979	3
1980-1984	13
1985-1989	32
1990-1994	79
1995-1999	188
>2000	88
Total general	405

Tabla 2.6 Clasificación de acuerdo al tipo de puesto ejercido.

# de Casos	
Nivel del Puesto	Total
ANALISTA	114
GERENTE	68
JEFATURA	45
COORDINADOR	38
ADMINISTRADOR	25
CONTADOR	24
CONSULTOR	11
CONTRALOR	8
DIRECTOR	7
AUDITOR	6
PROFESOR	5
OTROS	54
Total general	405

Tabla 2.7 Clasificación de acuerdo al área de desempeño profesional.

# de Casos	
Area del Puesto	Total
FINANZAS	107
ADMINISTRACION	74
SISTEMAS	73
INGENIERIA	40
VENTAS	28
CALIDAD	22
DOCENTE	18
RECURSOS HUMANOS	16
COMPRAS	9
OTROS	18
Total general	405

Ahora verificamos el valor de “p” que había sido estimado al 5% por el postulante en el apartado 2.5.a. A continuación se muestra la incidencia de opinión de los profesionistas respecto a la frecuencia con la cual ellos califican la aplicabilidad de las matemáticas en toma de decisiones en los negocios.

Tabla 2.8 Incidencia porcentual de la aplicación de las matemáticas en la toma de decisiones.

Frecuencia	Porcentaje
NUNCA	44
RARA VEZ	33
A VECES	17
FRECUENTEMENTE	6
Total general	100

Como se puede apreciar solo el 6% de la muestra mencionó de manera “regular” aplicar las matemáticas. Por tal motivo la estimación de la muestra fue bastante acertada. No obstante como ya se verificó, el tamaño de la muestra que se realizó (405), sobrepasa por mucho las exigencias mínimas del muestreo requerido.

Antes que nada, cabe señalar que está fuera del alcance del presente estudio de caso el desarrollar un análisis que vengan a incluir como variables inferenciales todos los elementos descriptivos mencionados en las tablas 2.2 al 2.7. La prueba “chi cuadrada” y el análisis de tablas de contingencia podrían ser utilizados para este otro propósito. Centrándonos ahora ya en la parte de las variables recabadas en el instrumento podemos describir la siguiente información contenida en las tablas 9 al 11.

Tabla 2.9 Incidencia del grado de importancia atribuida a cada variable.

	EN DESACUERDO	RARA VEZ INFLUYE	INDIFERENTE	A SER TOMADO EN CUENTA	TIENE MUCHA IMPORTANCIA	ES LA CAUSA RAIZ
CONCEPTOS	0	1	2	3	4	5
VAR.25	39	36	44	68	97	121
VAR.4	26	44	52	61	127	95
VAR.28	25	45	63	71	116	85
VAR.3	38	45	70	92	83	77
VAR.24	39	52	53	76	112	73
VAR.16	48	60	67	74	95	61
VAR.34	37	53	101	78	81	55
VAR.37	26	54	74	99	97	55
VAR.8	33	57	80	90	94	51
VAR.18	27	40	76	114	97	51
.....
VAR.n	An	Bn	Cn	Dn	En	Fn

La Tabla 2.9 muestra la incidencia de nivel 5 (causa raíz) para cada variable en forma descendente. Por ejemplo, se identifican 121 observaciones que consideran que la variable No. 25 (falta de habilidad para el razonamiento lógico de los problemas por parte del profesionista) como una variable que es causa raíz del problema. No obstante como se alcanza a apreciar en la misma tabla, existen otras observaciones que no piensan igual ya que, por ejemplo, hay 39 de ellas que están en completo de acuerdo, esto representa el grado de variabilidad de la variable, valga la redundancia. Más aún, existen hacia abajo otras variables que quizás no tengan la misma incidencia en el nivel 5, pero que si tengan menor variabilidad en sus observaciones registradas. Con esto se puede comprender que la estadística descriptiva tiene sus limitaciones a la hora de requerir generar conclusiones sobre las muestras. Lo anterior se logra resolver mediante la aplicación de la estadística inferencial, lo cual es precisamente lo que haremos.

La tabla 2.10 muestra de nuevo la lista de las variables. En este caso se vuelven a poner en forma descendente de acuerdo al nivel de importancia 5, pero adicionalmente se identifican que no todas las variables que aparecen en forma descendente corresponden al mismo constructo. Esto evidencia que la opinión de los encuestados no apunta a establecer un solo constructo como el único causante del problema.

Tabla 2.10 Incidencia del grado de importancia nivel “5” atribuido a cada variable.

VARIABLES	CONSTRUCTO	SIGNIFICADO DE LA VARIABLE	FREC. GRADO 5
VAR.25	EL ALUMNO	FALTA DE HABILIDAD PARA EL RAZONAMIENTO LOGICO DE LOS PROBLEMAS	121
VAR.04	EL MAESTRO	FALTA DE HABILIDAD PARA EL RAZONAMIENTO LOGICO DE LOS PROBLEMAS	95
VAR.28	EL METODO DE ENSEÑANZA	ENSEÑANZA MECANIZADA CON FORMULAS Y PROCEDIMIENTOS	85
VAR.03	CULTURA NACIONAL	MENOSPRECIO AL USO METODOS FORMALES	77
VAR.24	LA EMPRESA	FALTA DE CONOCIMIENTO PARA SU APLICACIÓN	73
VAR.16	CULTURA NACIONAL	CULTURA TERCER-MUNDISTA	61
VAR.37	EL ALUMNO	FALTA DE DISCIPLINA	55
VAR.34	EL METODO DE ENSEÑANZA	FALTA DE VINCULACIÓN CON LAS NECESIDADES DE LAS EMPRESAS	55
VAR.18	CULTURA NACIONAL	FALTA DE CULTURA COMPETITIVA	51
VAR.08	LA EMPRESA	FALTA DE TIEMPO (TOMA DE DECISIONES URGENTES)	51
VAR.50	ANTECEDENTES Y ESPECTATIVAS	ES MAS FACIL APLICAR EL SENTIDO COMÚN QUE LAS MATEMÁTICAS	50
VAR.23	LA EMPRESA	FALTA DE INTERES PARA SU APLICACIÓN	50
VAR.29	EL METODO DE ENSEÑANZA	FALTA DE PROMOCION A DESARROLLO DE HABILIDADES COMPETITIVAS	48
VAR.36	EL ALUMNO	FALTA DE INTERES Y VOCACION	46
VAR.27	EL ALUMNO	FALTA DE MOTIVACION	45
VAR.33	EL METODO DE ENSEÑANZA	PLANES DE ESTUDIO OBSOLETOS	44
VAR.12	EL MAESTRO	FALTA DE ACTUALIZACION Y/O EXPERIENCIA	38
VAR.38	EL METODO DE ENSEÑANZA	FALTA DE PERTINENCIA EN LA SOLUCIÓN DE PROBLEMAS REALES	38
VAR.13	EL MAESTRO	MAESTROS MAL PREPARADOS PEDAGOGICA-MENTE	37
VAR.46	ANTECEDENTES Y ESPECTATIVAS	EN MI CARRERA NO APRENDÍ LA FORMA DE APLICAR LAS MATEMÁTICAS	36
VAR.32	EL ALUMNO	CONFORMISMO	35
VAR.21	EL MAESTRO	FALTA DE MOTIVACIÓN	34
....
VAR.n	CONSTRUCTOx	SIGNIFICADO.X	FREC.X

2.5.c Exposición de resultados y comentarios finales.

Nuestro objetivo es identificar cuáles son los factores que a juicio de los profesionistas, mayormente explican el porqué no se aplican las matemáticas en la toma de decisiones en la administración. Para ello se hizo uso de la Prueba “t” de Student. A continuación se explica a grandes rasgos la metodología utilizada.

Se desarrolló un estudio por separado para cada uno de los 8 constructos ya antes explicados los cuales agrupan las 64 variables. Como es de esperarse cada constructo entonces estará compuesto a su vez por 8 variables. Para las 8 variables de cada constructo y a su vez para los 8 constructos, se obtuvo su valor “t” calculado, el cual luego fue confrontado contra el valor “t” de tablas para determinar la significancia y el porcentaje de variabilidad explicado para cada variable en cuestión. En la siguiente Tabla 2.11 se muestran algunos resultados al respecto.

Tabla 2.11 Porcentajes de variabilidad explicada para cada variable.

CONSTRUCTO	VARIABLE	% de Variabilidad Explicada
Maestro	FALTA DE HABILIDAD PARA EL RAZONAMIENTO LOGICO DE LOS PROBLEMAS	80%
Alumno	FALTA DE HABILIDAD PARA EL RAZONAMIENTO LOGICO DE LOS PROBLEMAS	80%
Maestro	FALTA DE ACTUALIZACION Y/O EXPERIENCIA	75%
Método	ENSEÑANZA MECÁNIZADA CON FORMULAS Y PROCEDIMIENTOS	74%
Maestro	FALTA DE CULTURA COMPETITIVA	72%
Maestro	FALTA DE DISCIPLINA	70%
Maestro	FALTA DE INTERES Y VOCACION	69%
Maestro	FALTA DE MOTIVACIÓN	69%
Método	FALTA DE PROMOCION A DESARROLLO DE HABILIDADES COMPETITIVAS	68%
Alumno	FALTA DE INTERES Y VOCACION	66%
...
Constructo x	Variable x	% x

Para cada constructo de manera independiente se calculó su coeficiente Croanbach para determinar la confiabilidad del instrumento de medición. Como es de esperarse no todos los constructos resultaron ser significativos ni tampoco igualmente confiables. En la Tabla 2.12 se resumen los resultados obtenidos.

Tabla 2.12 Porcentajes de variabilidad explicada y confiabilidad para cada constructo.

CONCEPTO	<i>% de Variabilidad Explicada</i>	<i>% de Confiabilidad</i>
	<i>Total</i>	<i>Alfa Cronbach</i>
Maestro	70%	92%
Método	59%	93%
Empresa	55%	93%
Alumno	46%	87%
Cultura Nacional	43%	93%
Recursos	28%	93%
Matemáticas	20%	96%
Antecedentes-Espectativa	16%	96%

En la tabla anterior se aprecia que aunque los porcentajes de confiabilidad para cada constructo son bastante razonables, ninguno de ellos por sí solo logra explicar al menos en un 80% la causa del problema. Por ejemplo, de la tabla anterior, se puede decir muy confiablemente que el constructo “maestro” juega un papel importante en la razón del porque no se aplican las matemáticas en la toma de decisiones en las empresas. No obstante, también se puede afirmar con mucha certeza que no existe evidencia de que el constructo “Antecedentes y Expectativas” sea la causante del problema en cuestión.

La propuesta final del análisis estadístico ofrece la posibilidad de desarrollar un constructo compuesto, es decir, un constructo artificial formado únicamente por aquellas variables que mejor expliquen la causa del problema. Así pues, es de esperarse que este constructo compuesto esté conformado por variables provenientes de los diferentes 8 constructos originalmente planteados en el instrumento de medición. Para esto se seleccionaron las primeras 10 variables que mejor desempeño tienen al dar explicación al problema. Los resultados obtenidos al respecto son muy favorables tanto en el porcentaje de variabilidad explicada del constructo así como en su confiabilidad. En la Tabla 2.13 se muestran los resultados.

Tabla 2.13 Porcentajes de variabilidad explicada para las variables del constructo compuesto.

CONSTRUCTO	VARIABLE	% de Variabilidad Explicada
Maestro	FALTA DE HABILIDAD PARA EL RAZONAMIENTO LOGICO DE LOS PROBLEMAS	90%
Alumno	FALTA DE HABILIDAD PARA EL RAZONAMIENTO LOGICO DE LOS PROBLEMAS	86%
Maestro	FALTA DE ACTUALIZACION Y/O EXPERIENCIA	89%
Método	ENSEÑANZA MECÁNIZADA CON FORMULAS Y PROCEDIMIENTOS	76%
Maestro	FALTA DE CULTURA COMPETITIVA	82%
Maestro	FALTA DE DISCIPLINA	69%
Maestro	FALTA DE INTERES Y VOCACION	77%
Maestro	FALTA DE MOTIVACIÓN	76%
Método	FALTA DE PROMOCION A DESARROLLO DE HABILIDADES COMPETITIVAS	66%
Alumno	FALTA DE INTERES Y VOCACION	63%

Como puede apreciarse en la tabla anterior, el desempeño de las variables ahora ya interrelacionadas mejora aún más en relación a su desempeño al haber estado en sus constructos originales. El coeficiente de confiabilidad Croanbach para este nuevo constructo tiende a disminuir por la diversidad de las variables que lo componen, pero con un 83% que resulta de su calculo vía el Software para Análisis Estadístico SPSS, aún continúa siendo bastante aceptable para nuestros fines.

Finalizamos esta sección diciendo que en el desarrollo del último constructo compuesto no todas las variables tuvieron el mismo nivel de desempeño, aunque tampoco era lo esperado. De estas diferencias en su desempeño es de donde logramos establecer 4 factores que resultan concluyentes del estudio de campo:

1. Existe una falta de habilidad para el razonamiento lógico de los problemas tanto por parte del maestro como por parte del alumno.
2. Existe una falta de actualización y experiencia por parte del maestro.
3. Existe una falta de cultura orientada hacia la competitividad.
4. Existe una deficiencia en el método de enseñanza de las matemáticas derivada del uso mecanizado de formulas.

El estudio de campo desarrollado y expuesto muestra con gran claridad que los antecedentes y aspectos teóricos discutidos previamente tienen relevancia y pertinencia de acuerdo al común de las opiniones recabadas en dicho experimento. Las evidencias estadísticas ya antes expuestas así lo prueban.

2.6 Comentarios finales de la problemática educativa en la aplicación del OR/MS.

A partir de los resultados del “estudio de campo” presentados en el apartado 2.5, la opinión del postulante es que la experiencia y la extrapolación del pasado sigue siendo la principal herramienta para la toma de decisiones actualmente en las organizaciones. En la práctica de los negocios, pocas veces ocurre que existan modelos que logren incluir las variables y las relaciones mediante las cuales el administrador pueda mantener sensitivamente tanto el control de la situación entrante, así como también el entendimiento de las acciones sugeridas respecto a la problemática a resolver. No obstante que la investigación pura es necesaria, existe la preocupación de que sólo una muy pequeña contribución del OR/MS se aplique hoy en día en la administración de las empresas en general y en la logística en particular.

En la Figura 2.3 se expone un diagrama esquemático con la finalidad de resaltar los conceptos revisados.

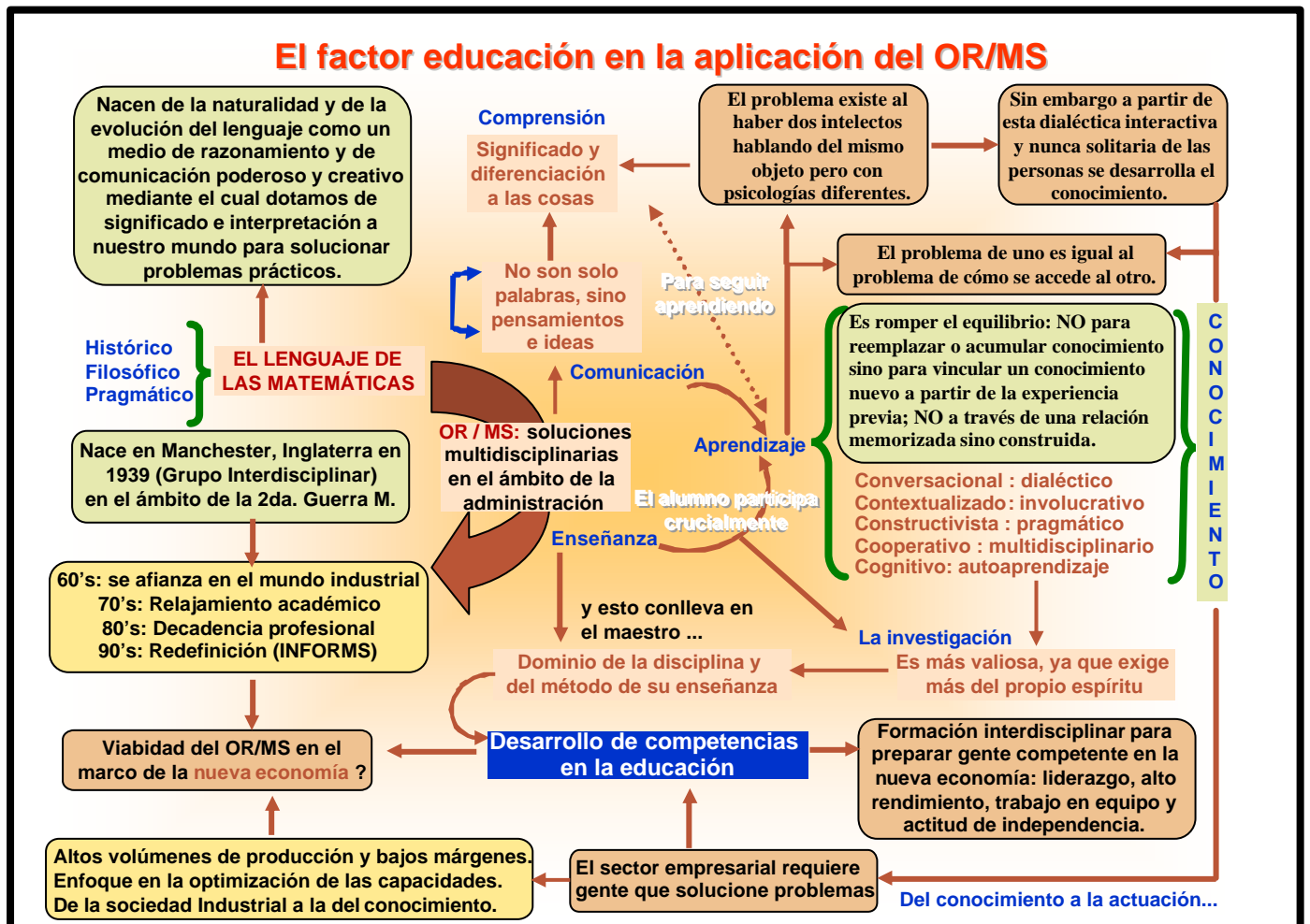


Figura 2.3 Diagrama esquemático acerca del factor educación en la aplicación del OR/MS.

La relación específica que guarda el tema del desarrollo de “competencias” en la educación en México con respecto a la tesis doctoral se refiere a la necesidad de establecer antes que nada un supuesto básico que debe existir en toda sociedad que pretenda dar solución a problemas como el que será atendido en la presente tesis doctoral. Es decir, antes que un algoritmo o propuesta matemática, lo primero que debe existir para dar solución a un problema, es la presencia de un individuo (tomador de decisiones) que atienda el problema como tal, lo comprenda en términos de un modelo conceptual, relacional y lógico; logrando identificar la información que sea relevante. Cubierto el supuesto anterior, luego entonces ahora sí podremos pasar al tema de las matemáticas y de los algoritmos. Dicho de otro modo, el punto de partida está en el modelo no en el algoritmo.

De la argumentación anterior, es fácilmente comprensible que el problema particular de la educación y las “competencias” antes descrito es por sí solo materia suficiente para un estudio por separado. Aunque por tal motivo, este tema particular escapa al alcance de investigación de la presente tesis doctoral, nos permitimos en el último punto del presente capítulo realizar un estudio de campo que pretendió estudiar un poco más de cerca dicho fenómeno. Con esto damos por finalizado el presente tema para dirigirnos en el capítulo 3 a la exposición de algunos problemas que resultan ser típicos a ser enfrentados en la práctica de la logística e iniciar de este modo formalmente el desarrollo del marco conceptual de la tesis.

De los problemas que se expongan en el capítulo 3, uno de ellos será seleccionado como tema de investigación (problema de logística de ruteo de distribución). Luego entonces en el capítulo 4 se dará tratamiento formal a los fundamentos matemáticos de los problemas generales de ruteo de distribución. A continuación en el capítulo 5 se presentarán formalmente los fundamentos teóricos aplicables a la propuesta meta-heurística que el postulante expondrá como hipótesis para la solución del problema de investigación. Luego entonces, dicha propuesta meta-heurística estará basada en la aplicación de un algoritmo genético para resolver un problema de logística de ruteo de distribución el cual se planteará formalmente y de manera específica, con sus características y propiedades, en el capítulo 6.

Por razones que obedecen al balance en el contenido del presente documento, todos los aspectos referidos a los objetivos, la importancia y las delimitaciones que será tomados en cuenta para caracterizar al problema de investigación doctoral, se abordarán entonces en el capítulo 7 de la tesis. El diseño experimental de la investigación se expondrá en el capítulo 8 y finalmente la presentación de los resultados y las conclusiones se desarrollarán en los capítulos 9 y 10 respectivamente.

3. Exposición de problemas generales en la logística.

3.1 Introducción.

En la logística existen diversos problemas que comúnmente se presentan. A continuación en la Figura 3.1 se muestra un diagrama que describe 6 de estos problemas que con frecuencia se presentan en la práctica.

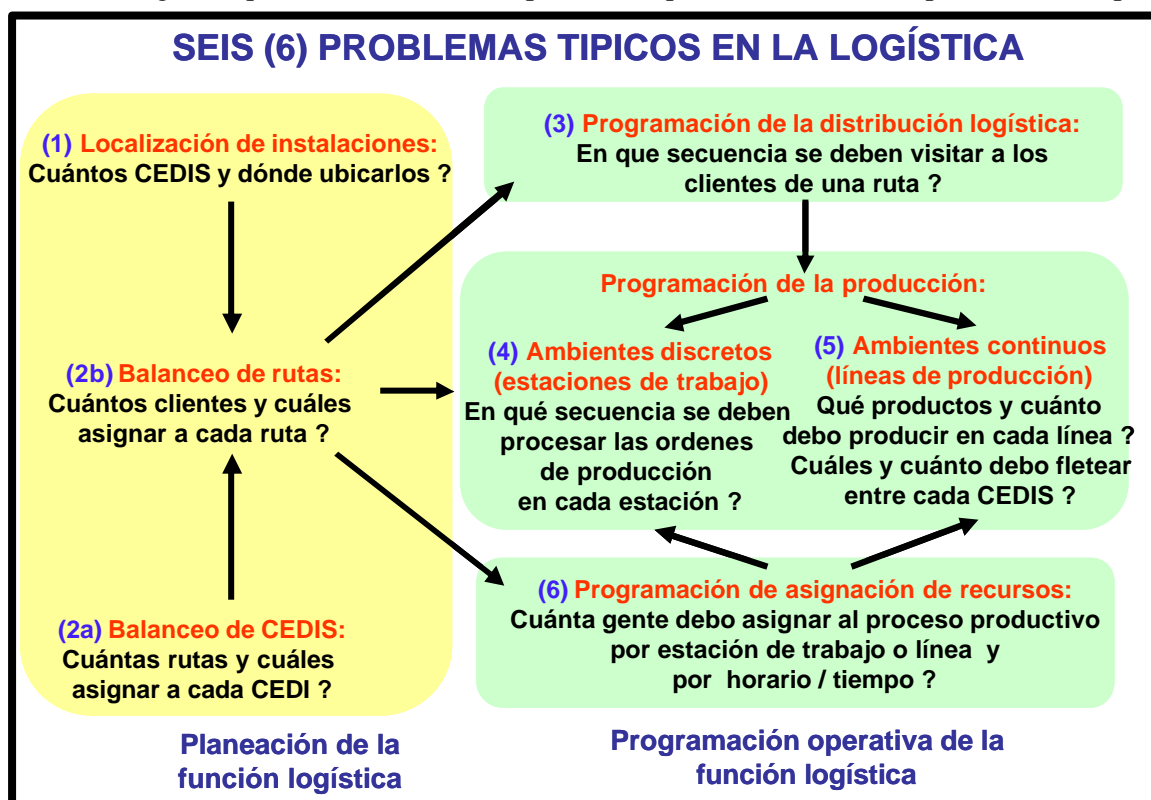


Figura 3.1 Seis problemas generales en la práctica de la logística.

De los 6 problemas expuestos en la figura anterior, a continuación se desarrolla una exposición general de cuatro de éstos. Estos 4 problemas guardan una relación en común la cual se expondrá en el apartado 3.6. La intención de exponer estos 4 problemas es con la finalidad de delimitar cuál será tomado como el problema a desarrollar en el proyecto de investigación. El último apartado corresponde a un estudio de caso mediante el cual se tiene el objetivo de identificar cuál es la oferta de software comercial en la ciudad de Monterrey que se tiene para buscar resolver este tipo de problemas en cuestión. Aunque dicho estudio se desarrolló en mayo del 2002, los resultados previsiblemente siguen siendo válidos aún en la actualidad.

3.2 Primer problema de la logística: planeación estratégica de red e infraestructura.

En las empresas de distribución siempre será crítico establecer la ubicación adecuada para todas aquellas instalaciones que darán operatividad al negocio. Aquí nos estamos refiriendo concretamente a las decisiones que tienen que ver con localización de plantas, centros de distribución (CEDIS), nodos de transferencia, bodegas, sucursales, etc. Las decisiones referentes a dónde ubicar este tipo de instalaciones tienen que considerar factores de carácter cualitativo tales como:

1. Viabilidad y tránsito de la zona.
2. Regulaciones en materia de urbanización y uso del suelo.
3. Restricciones referentes a manejo de materiales peligrosos.
4. Plusvalía del terreno.
5. Poder adquisitivo de la zona.
6. Areas potenciales de mercado, crecimientos poblacionales y tendencias de mercado para cada una de las zonas a ser atendidas por los nodos.
7. La conveniencia económica y estratégica que pueda obtenerse en materia de distribución logística una vez que se logre tener el producto más cerca del cliente.

Es importante puntualizar que tener el producto más cerca del cliente y operar con una logística de distribución más ágil, genera una barrera de entrada al resto de los competidores. Así pues, las medidas estratégicas que puedan formularse en términos de un incremento en la efectividad logística, contribuirán en la ventaja competitiva del negocio.

Podemos iniciar buscando contestar a la interrogante acerca de cuál es la cantidad óptima de instalaciones, plantas o CEDIS con la cual el negocio debe operar. Llamaremos de aquí en adelante a esto la cantidad óptima de nodos, los cuales pueden ser categorizados en nodos oferentes (plantas) o en nodos demandantes (almacenes, sucursales) de acuerdo a la naturaleza de cada uno de éstos.

La dificultad aparece una vez que se comparan los costos de distribución versus los costos fijos de operación de los nodos a ser empleadas para atender la red logística. Es decir, al incrementar la cantidad de nodos que operan en el sistema, es cierto que disminuyen los costos de distribución ya que los productos a ser distribuidos hacia los clientes finales estarán mucho más cerca; pero también entonces es necesario advertir que los costos de operación se incrementarán a consecuencia de tener que operar con una cantidad

mayor de nodos. Entonces el objetivo aquí es encontrar aquella cantidad óptima de nodos que haga que la combinación de ambos costos, es decir, los de distribución que son los costos variables y los de operación en los nodos que son los fijos, ambos sumados den el menor costo total de la red logística.²⁹

Cubierta la problemática anterior, luego se añade la dificultad de definir en qué ubicación serán instalados cada uno de los nodos que darán operación a la red de distribución logística. Sin hacer a un lado todas las consideraciones de origen práctico que ya antes se explicaron a este respecto, de una manera conceptual, podemos definir que la ubicación óptima de los nodos estará definida por aquella combinación de ubicaciones que mejor repartan la carga logística y geográfica que cada uno de los nodos tendrá que soportar en términos de distancias, densidades de población, niveles de servicio y políticas de cobertura de demanda a lo largo y ancho de un territorio definido.

Prosiguiendo en la definición de nuestra estrategia de distribución logística, el siguiente aspecto a contemplar corresponde a determinar la cantidad y la delimitación geográfica de aquellas áreas o rutas que serán asignadas a ser atendidas por cada nodo de acuerdo a la ubicación óptima que haya resultado en el razonamiento anterior. Contestar esta interrogante, nos lleva a tener que considerar restricciones de diversa índole tales como las siguientes:

1. Capacidad de almacenamiento, capacidad de flujo entrada y de flujo de salida para cada uno de los nodos de la red.
2. Nivel de equipamiento en cada nodo (montacargas, andenes, equipo para manejo de materiales, transportadores, etc) que servirán para la atención operacional de las rutas a poder ser asignadas.
3. Capacidad de auto-sustentabilidad para cada uno de los nodos para asegurar el servicio y el mantenimiento de la cantidad y el tipo de rutas a ser asignadas (taller mecánico, instalaciones para despacho de combustible, oficinas administrativas, etc.)
4. Consideración de los límites naturales o topográficos que pudieran ser mejor aprovechados para delimitar el territorio de cobertura correspondiente a las rutas a ser atendidas por cada nodo.
5. Nivel de densidad de volumen de demanda y de clientes de los territorios que fuesen a ser asignados a cada CEDIS.

Nuevamente aquí, en términos conceptuales hablaríamos de encontrar aquella combinación que mejor reparta los territorios de las rutas entre cada uno de los nodos que vayan a estar en operación en la red

²⁹ Drezner, Zvi. (1996), *Facility Location: A Survey of Applications and Methods*, 1st Ed, Springer Verlag, EUA, pag 130-167.

logística. Cuando mencionamos "mejor asignación de los territorios de las rutas", nos estamos refiriendo a aquella combinación que se busca y en la cual obtuviéramos una carga de trabajo razonablemente balanceada entre los nodos de acuerdo a las consideraciones de capacidad y de densidad de demanda antes mencionadas.

Finalmente llegamos al nivel atómico de nuestro plan estratégico logístico. Es aquí cuando debemos determinar la cantidad y la ubicación geográfica de los clientes que serán atendidos por cada una de las rutas que fueron asignadas ya anteriormente a cada uno de los nodos y que también éstos hayan resultado ser los óptimos tanto en cantidad como en ubicación geográfica.

Establecer los clientes a ser asignados a cada ruta conlleva determinar cuáles y por ende cuántos de estos clientes serán atendidos por cada una de las rutas y esto a su vez nos lleva a tomar en cuenta las siguientes consideraciones:

1. Densidad de la zona en términos de la cantidad de clientes actuales y potenciales.
2. Nivel de demanda en volumen de cada uno de los clientes dentro del territorio.
3. Clasificación y estratificación de los tipos de clientes de acuerdo a la naturaleza del canal comercial que mejor pueda atender.
4. Capacidades de carga (camión, camioneta, automóvil, motocicleta, trailer, etc.) que podrá ser empleada para hacer llegar la distribución física al cliente.
5. Restricciones de vialidad de tránsito de la zona y aspectos de índole topográfico del terreno.
6. Aspectos del tipo cualitativo para la asignación de clientes a un vendedor tales como empatía, concesiones, negociaciones, sindicato, etc)

De igual forma, aquí también en términos conceptuales hablaríamos de encontrar aquella repartición y cantidad óptima de clientes a ser asignada en cada ruta logrando que mejor reparta y balancee la carga de trabajo y el nivel de servicio entre las rutas de acuerdo a las consideraciones antes mencionadas. A continuación se presenta el modelo matemático para la solución del problema en lo referente a la fase de determinación de los clientes que agruparan a cada una de las rutas con la finalidad de balancear las cargas de trabajo entre ellas.

Balanceo de carga de ruta Multicriterio con Elegibilidad (Extensible Two-Dimensional BPP with Eligibility)

Variables de Entrada (1 de 2):

*Dominios: $i, j = 1 \dots n$ (Manzanas de un territorio a ser agrupadas)
 $k = 1 \dots m$ (Rutas a formar en el territorio)*

T_{ij} = Cantidad de clientes agrupados en las manzanas "i" + "j"

V_{ij} = Volumen de ventas agrupado para las manzanas "i" + "j"

En ambos casos el valor de la variable tipo "i,j" es la suma aritmética de los clientes y del volumen de ventas agrupado en ambas manzanas "i,j".

Para el caso donde "i=j" (diagonal), entonces se suma UNA SOLA VEZ.

A_{ij} = Afinidad de la manzana "i" con la manzana "j"

$\begin{cases} 0 = \text{La manzana "i" PUEDE ser agrupada con la manzana "j"} \\ 1 = \text{La manzana "i" DEBE ser agrupada con la manzana "j"} \end{cases}$

B_{ij} = Afinidad de la manzana "i" con la manzana "j"

$\begin{cases} 0 = \text{La manzana "i" PUEDE ser agrupada con la manzana "j"} \\ 1 = \text{La manzana "i" NO DEBE ser agrupada con la manzana "j"} \end{cases}$

Variables de Entrada (2 de 2):

C_{ij} = Costo / Colindancia para agrupar la manzana "i" con la manzana "j"

*$\begin{cases} 0 = \text{La manzana "i" es colindante con la manzana "j"} \\ \infty = \text{La manzana "i" NO es colindante con la manzana "j"} \end{cases}$
Donde $C_{ii} = 0$*

CL_k = Cantidad mínima de clientes a ser asignados en la ruta "k"

CU_k = Cantidad máxima de clientes a ser asignados en la ruta "k"

VL_k = Volumen mínimo de ventas a asignar en la ruta "k"

VU_k = Volumen máximo de ventas a asignar en la ruta "k"

Variables de Salida (1 de 2):

$$X_{ijk} \geq 0, \leq 1, \text{ent} \begin{cases} 0 \rightarrow \text{El cluster "i", "j" NO se va a asignar a la ruta "k"} \\ 1 \rightarrow \text{SI} \end{cases}$$

Donde : $i, j = 1 \dots n$ (Manzanas a ser agrupadas)
 $k = 1 \dots m$ (Rutas a ser formadas)

Dado que la Matriz C_{ij} es Simétrica entonces: $X_{ijk} = X_{jik}$, para: $j \geq i$
Además es posible explotar la estructura del problema dejando de utilizar una gran cantidad de variables binarias que NO son viables debido a las colindancias topológicas de la red...

Por ejemplo: si tuviéramos 1000 Manzanas a ser agrupadas en 100 Rutas y usualmente existieran 6 colindancias en promedio para cada manzana:

$$X_{ijk} \text{ Canónico: } 1000 * 1000 * 100 = 100,000,000 \text{ var's}$$

$$X_{ijk} \text{ Simétrico: } \text{aprox } 1000 * 1000 * 100 / 2 = 50,500,000 \text{ var's}$$

$$X_{ijk} \text{ Explotado: } \text{aprox } 1000 * 6 * 100 / 2 = 300,000 \text{ var's}$$

No obstante el espacio de búsqueda sigue siendo enorme, podemos afirmar que el problema empieza a parecer "tratable" computacionalmente hablando.

Variables de Salida (2 de 2):

$D_c (\geq 0)$: Tolerancia porcentual negativa referida al # de clientes por ruta.

$S_c (\geq 0)$: Tolerancia porcentual positiva referida al # de clientes por ruta.

$D_v (\geq 0)$: Tolerancia porcentual negativa referida al volumen de ventas por ruta.

$S_v (\geq 0)$: Tolerancia porcentual positiva referida al volumen de ventas por ruta.

Restricciones (1 de 5):

Cada manzana debe estar asignada en 2 arcos (i,j):

- * Para las manzanas primera y última en ser agrupadas a la ruta se tendrá $X_{iik} = 1$ y además el arco $X_{ijk} = 1$ que sirva para ligar la 1era manzana con la 2da y la penúltima con la última.
- * Para el resto de las manzanas, se tendrán 2 asignaciones, una ligada hacia la manzana anterior y otra hacia la manzana siguiente.

$$\sum_{k=1}^m \sum_{i=1}^h X_{ihk} + \sum_{k=1}^m \sum_{j=h+1}^n X_{hjk} = 2$$

para $h = 1..n$; y donde "i", "j" sean colindantes con "h"

Cada ruta debe tener solamente 2 asignaciones del tipo $X_{iik} = 1$ (Variables de la Diagonal):

- * Una para la manzana "origen" y
- * La otra para la manzana "destino".

\therefore La Ruta NO puede estar picoteada

$$\sum_{i=1}^n X_{iik} = 2$$

para $k = 1..m$

Restricciones (2 de 5):

$$\sum_{i=1}^h X_{ihk} + \sum_{i=1}^h X_{ihl} \leq 1 \qquad \sum_{j=h+1}^n X_{hjk} + \sum_{j=h+1}^n X_{hjl} \leq 1$$

{ Cada arco de manzanas "i, h" o "h, j" puede ser asignado a una sola ruta }

para $h = 1..n$;

$k = 1..m$;

$l = k + 1..m$

y donde "i", "j" sean colindantes con "h"

$$\sum_{i=1}^h X_{ihk} + \sum_{j=h+1}^n X_{hjl} \leq 1 \qquad \sum_{j=h+1}^n X_{hjk} + \sum_{i=1}^h X_{ihl} \leq 1$$

{ Cada manzana "h" puede ser asignada a una sola ruta }

para $h = 1..n$;

$k = 1..m$;

$l = k + 1..m$

y donde "i", "j" sean colindantes con "h"

Restricciones (3 de 5):

$$2VL_k^*(1-Dv) \leq \sum_{h=1}^n \sum_{i=1}^h X_{ihk} * V_{ih} \leq 2VU_k^*(1+Sv)$$

{ para cada ruta "k", el volúmen de ventas asignado debe cubrir el
mínimo necesario y no debe exceder del máximo (capacidad de camión). }

para $k = 1..m$

$$2CL_k^*(1-Dc) \leq \sum_{h=1}^n \sum_{i=1}^h X_{ihk} * T_{ih} \leq 2CU_k^*(1+Sc)$$

{ para cada ruta "k", la cantidad de clientes asignados debe cubrir el
mínimo necesario y no debe exceder del máximo (capacidad de servicio). }

para $k = 1..m$

En las formulaciones a través de las variables T_{ih} (# de Clientes) y de
 V_{ih} (Volúmen de Ventas)

en ambos casos se requiere dividir entre 2, debido a que cada manzana
aparece exactamente en dos agrupaciones.

Restricciones (4 de 5):

Cuando $B_{hl} = 1$, entonces solo cuando mucho 2 arcos podrán estar asignados a la ruta "k",
ya sea correspondan a la manzana "h" o a la manzana "l", pero no ambas manzanas
podrán estar asignadas para la misma ruta "k".

$$\left(\sum_{i=1}^h X_{ihk} + \sum_{j=h+1}^n X_{hjk} \right) + \left(\sum_{i=1}^l X_{ilk} + \sum_{j=l+1}^n X_{ljk} \right) \leq \overline{LI} - \overline{LS}$$

$$0 \quad 4 \quad B_{hl} = 0$$

para $k = 1..m$; $h = 1..n$; $l = h+1..n$

$$0 \quad 2 \quad B_{hl} = 1$$

considerando que las manzanas "i", "j" sean colindantes con "h" y con "l"

$$4(1-B_{hl}) + 2B_{hl} \geq \left(\sum_{i=1}^h X_{ihk} + \sum_{j=h+1}^n X_{hjk} \right) + \left(\sum_{i=1}^l X_{ilk} + \sum_{j=l+1}^n X_{ljk} \right)$$

simplificando...

$$\left(\sum_{i=1}^h X_{ihk} + \sum_{j=h+1}^n X_{hjk} \right) + \left(\sum_{i=1}^l X_{ilk} + \sum_{j=l+1}^n X_{ljk} \right) \leq 4 - 2B_{hl}$$

Restricciones (5 de 5):

Cuando $A_{hl} = 1$; entonces se restringe a que la diferencia de las 2 expresiones sea igual a 0

∴ Esto equivale a que cuando $A_{hl} = 1$ solo existan 2 posibilidades :

1. Ambas manzanas "h", "l" se asignan a la ruta "k" o,

2. Ninguna de las 2 manzanas se asigna a la ruta "k".

$$\left(\sum_{i=1}^h X_{ihk} + \sum_{j=h+1}^n X_{hjk} \right) - \left(\sum_{i=1}^l X_{ilk} + \sum_{j=l+1}^n X_{ljk} \right) \quad \overline{LI - LS}$$

$$\begin{matrix} -2 & +2 & A_{hl} = 0 \\ 0 & 0 & A_{hl} = 1 \end{matrix}$$

para $k = 1..m$; $h = 1..n$; $l = h+1..n$

considerando donde las manzanas "i", "j" sean colindantes con "h" y con "l"

$$-2(1-A_{hl}) \leq \left(\sum_{i=1}^h X_{ihk} + \sum_{j=h+1}^n X_{hjk} \right) - \left(\sum_{i=1}^l X_{ilk} + \sum_{j=l+1}^n X_{ljk} \right)$$

$$+2(1-A_{hl}) \geq \left(\sum_{i=1}^h X_{ihk} + \sum_{j=h+1}^n X_{hjk} \right) - \left(\sum_{i=1}^l X_{ilk} + \sum_{j=l+1}^n X_{ljk} \right) \quad \text{luego simplificando...}$$

$$\left(\sum_{i=1}^h X_{ihk} + \sum_{j=h+1}^n X_{hjk} \right) - \left(\sum_{i=1}^l X_{ilk} + \sum_{j=l+1}^n X_{ljk} \right) \geq 2A_{hl} - 2$$

$$\left(\sum_{i=1}^h X_{ihk} + \sum_{j=h+1}^n X_{hjk} \right) - \left(\sum_{i=1}^l X_{ilk} + \sum_{j=l+1}^n X_{ljk} \right) \leq 2 - 2A_{hl}$$

Función Objetivo:

$$FO_{min} : P_1 \left[\sum_{k=1}^m \sum_{h=1}^n \sum_{i=1}^h X_{ihk} * C_{ih} + \sum_{k=1}^m \sum_{h=1}^n \sum_{j=h+1}^n X_{hjk} * C_{hj} \right] +$$

$$P_2 (Dc + Sc) +$$

$$P_3 (Dv + Sv)$$

Donde las manzanas "i", "j" sean colindantes con la manzana "h"

P_1 = Nivel de importancia relativa para el objetivo de cumplir con las restricciones topológicas de colidancia en la red.

P_2 = Nivel de importancia relativa para el objetivo de cumplir con las restricciones de balanceo de clientes por ruta.

P_3 = Nivel de importancia relativa para el objetivo de cumplir con las restricciones de balanceo de volumen de ventas por ruta.

Después de presentar la modelación matemática de cada problema general de la logística, haremos una breve descripción de dicho modelo con el propósito de finalizar la exposición de cada caso en particular. La modelación antes expuesta corresponde a nuestro problema de planificación estratégica de la red logística. En términos taxonómicos el modelo del problema puede ser categorizado como uno de balanceo de carga de ruta multi-criterio con elegibilidad. El problema anterior corresponde a los relacionados al tema de “empacamiento” (o packing en inglés), por tanto el problema corresponde al tipo de problemas difíciles de resolver (o NP-Hard en inglés), de este último tema se hablará más a detalle en el capítulo 5.

El modelo anterior desarrolla algunos conceptos que generalmente son comunes encontrar en el problema de planificación de rutas como son los siguientes:

1. Cantidad de clientes que deben ser atendidos en cada ruta.
2. Volumen de ventas que debe ser abastecido en cada ruta.
3. Criterios de tolerancia permitidas para el balanceo de cargas entre las rutas.

Adicionalmente a lo anterior, el modelo particularmente dedica gran parte de su estructura matemática a atender la afinidad que existe en el agrupamiento de las manzanas que van a formar los territorios de las rutas. El asunto de la afinidad de las manzanas cubre los siguientes aspectos:

1. Las manzanas a ser agrupadas deben guardar una continuidad o colindancia geográfica en el territorio. Lo anterior es con el objetivo de asegurar que los territorios no estén “picoteados”, es decir manzanas que no estén contiguas dentro del territorio.
2. El modelo adicionalmente cubre el caso en el que aunque las manzanas si son contiguas, por razones diversas el planeador desea que no queden agrupadas en la misma ruta.
3. Coincidentemente también el modelo cubre aquel caso en el que el planeador desea el que dos o más manzanas queden agrupadas en una misma ruta.

En general, el modelo dedica mucho de su desarrollo matemático a explotar la estructura del problema. Prueba de ello es que la variable X_{ijk} es definida en términos de hacer uso de la simetría que existe al agrupar cada par de manzanas. Es decir, resulta conveniente aprovechar el hecho de que las propiedades de planificación que se logran al agrupar una primera manzana con una segunda resultan equivalentes a que cuando se hacen en sentido inverso.

Adicional a lo anterior, el modelo propone como parte de una fase de pre-proceso algorítmico, una heurística bastante conveniente mediante la cual es posible establecer aquellas agrupaciones de manzanas

que resultan ser infactibles debido a la lejanía geográfica relativa que pueda haber entre ellas. La heurística anterior resulta bastante ventajosa ya que como se expone en la modelación, reduce enormemente la cantidad de variables y por consiguiente el tamaño del espacio solución que el algoritmo tuviera que explorar para la búsqueda de la solución óptima.

Cabe puntualizar que al ser definida simétricamente la variable " X_{ijk} ", se tiene el beneficio de reducir el espacio solución del problema, pero también es importante mencionar que el precio a pagar de dicha ventaja corresponde a la necesidad de desarrollar un sistema de restricciones más complejo para atender dicha estructura. Dicha situación puede ser verificada en la modelación antes expuesta.

3.3 Segundo problema de la logística: programación de la producción para ambientes continuos.

La programación de la producción es uno de los problemas más importantes a resolver en la práctica actual de la logística de los negocios. Generalmente las empresas resuelven este problema sobre la base de la experiencia de la gente encargada de realizar dichas funciones. De hecho, es raro encontrar en las empresas que el área encargada de programar la producción sea la misma área que tiene la responsabilidad de programar la distribución inter-plantas e inter-cedis.

Posiblemente la principal razón de lo anterior es debido a que ya por sí solas cada una de estas funciones son lo suficientemente complejas como para lograr integrar en un solo sistema de trabajo toda la logística de suministro. No obstante, intrínsecamente es necesaria dicha integración a fin de asegurar una coordinada y ágil operación que asegure la optimización de la cadena de suministro. En principio, la programación de la producción debe tomar en cuenta las capacidades de producción presentes de la planta. Esto se refiere, por ejemplo, a aspectos tales como ³⁰:

1. Disponibilidad de cuadrillas de operadores.
2. Consideración y apego a los programas de mantenimiento preventivo.
3. Disponibilidad de inventarios de materias primas para el proceso productivo.
4. Restricciones referidos al tamaño de los lotes prácticos de producción.
5. Tiempos requeridos y costos implicados para los cambios de producto (Setup) en las líneas de producción.

De igual modo, la operación de distribución inter-plantas o inter-cedis conlleva la consideración de aspectos tales como:

1. Capacidades de almacenamiento en cada planta y CEDIS.
2. Capacidades de servicio y atención a los fleteos de entrada y de salida desde o hacia cada planta o CEDIS.
3. Políticas de cobertura de inventarios por tipo de producto y para cada planta o CEDIS.

Como puede apreciarse, el problema es complejo dado que, mantener una política generosa en los inventarios asegura el abastecimiento y la disponibilidad del producto, sin embargo, incrementa los costos

³⁰ Klein, Robert. (1999), *Scheduling of Resource-Constrained Projects*, Kluwer Academic, EUA, pag 77.

financieros, las mermas y los desperdicios causados por los excesos de inventarios en la operación. Por otro lado, si acaso la política de inventarios fuese muy rigurosa, entonces los costos financieros de los inventarios tenderían a bajar, pero se incrementaría la probabilidad de que lleguen a existir problemas de abasto, disponibilidad y oferta de producto en el mercado afectando potencialmente el volumen de ventas.

Encontrar el justo medio para la problemática anterior es solo el inicio, ya que luego será necesario incluir en la disyuntiva anterior, la necesidad de realizar frecuentes cambios de producto en las líneas de fabricación. Esto último ciertamente puede disminuir los costos financieros de los inventarios, pero por otro lado incrementará los costos y los tiempos improductivos que se generan cada vez que es necesario hacer un cambio en las líneas de producción, es decir se incrementan la cantidad de cambios o setups en las plantas.

Finalmente, es obvio decir que esta logística de suministro debe estar desarrollada bajo una base dinámica en el tiempo, siempre procurando apegarse a las condiciones cambiantes de la demanda del mercado al nivel de producto y territorio de venta. Para ello, el uso de un pronóstico de demanda como información entrante al modelo matemático, resultará crucial para su implementación en la práctica de los negocios.

El objetivo en esta cadena de suministro será el lograr ejecutarla siempre asegurando la disponibilidad del producto correcto, en la cantidad correcta y en el momento oportuno al menor costo posible. Así pues un posible criterio de optimización podría ser aquel basado en minimizar la suma total de los costos de producción, más los costos financieros de los inventarios, más los costos de oportunidad debido a los faltantes de producto.

A continuación se anexa el modelo matemático planteado para el problema en cuestión.

Variables Exógenas asociadas al Inventario

I_{ij0} = Inventario actual del producto "i" en la planta "j"


W_{ijkl} = Inventario estimado en proceso para el producto "i" en la línea de producción "k" localizada en la planta "j" y en el turno "l"

D_{ijl} = Demanda esperada para el producto "i" en la planta "j" en el turno "l"

P_{ij} = Política de inventarios mínimos para el producto "i" en la planta "j"

M_{ijl} = Inventario estimado de insumos para producir el producto "i" en la planta "j" en el turno "l"

F_{ij} = Política de inventarios de insumos mínimos requeridos

 para poder producir el producto "i" en la planta "j"

Variables Exógenas asociadas a las Capacidades de Producción

Q_{jl} = Cantidad de cuadrillas disponibles para producción en la planta "j" y en el turno "l"

R_{ik} = Lote práctico de producción para el producto "i" en la línea de producción "k"

Definición de disponibilidad de líneas de producción por turno :

$A_{kl} = \begin{cases} 0 & \text{No estará disponible la línea de producción "k" en el turno "l"} \\ 1 & \text{Si} \end{cases}$

Factibilidad del uso de las líneas de producción para cada producto

$B_{ik} = \begin{cases} 0 & \text{No es viable producir el producto "i" en la línea de producción "k"} \\ 1 & \text{Si} \end{cases}$

Variables Exógenas asociadas a las Capacidades en Almacenes

H_j = Capacidad de almacenaje para producto terminado en la planta " j "

E_{jl} = Capacidad de fleteo de salida de producto inter - plantas para la planta " j " en el turno " l "

S_{jl} = Capacidad de fleteo de entrada de producto inter - plantas para la planta " j " en el turno " l "

Variables Exógenas asociadas a los Costos de Operación

C_{ik} = Costo de producción unitario para el producto " i " en la línea de producción " k "

G_{jh} = Costo de fleteo unitario para enviar producto desde la planta " j " a la planta " h "

Variables Endógenas

$$X_{ijkl} \geq \begin{cases} 0 & \text{NO se produce el producto " i " en la línea de} \\ & \text{producción " k " de la planta " j " en el turno " l " } \\ 1 & \text{SI} \end{cases}$$
$$I_{ijl} \geq 0 \begin{cases} \text{Inventario final del producto " i " } \\ \text{en la planta " j " y en el turno " l " } \end{cases}$$
$$Y_{ijhl} \geq 0 \begin{cases} \text{Cantidad de unidades a fletear del producto " i " } \\ \text{en el turno " l " desde la planta " j " hacia la planta " h " } \end{cases}$$

Restricciones (1 de 3)

$$\sum_{j=1}^{n2} \sum_{i=1}^{n1} X_{ijkl} \leq A_{kl}$$

Para $k=1 \dots n3$
 $l=1 \dots n4$

Para cada línea de producción "k" y en cada turno "l", la decisión de producir un producto "i" en una planta "j" deberá considerar la disponibilidad de la línea.

$$\sum_{j=1}^{n2} \sum_{l=1}^{n4} X_{ijkl} \leq B_{ik}$$

Para $i=1 \dots n1$
 $k=1 \dots n3$

Para cada producto "i" y en cada línea de producción "k" la decisión de producir en un turno "l" y en una planta "j" deberá considerar su factibilidad.

$$\sum_{i=1}^{n1} \sum_{k=1}^{n3} X_{ijkl} \leq Q_{jl}$$

Para $j=1 \dots n2$
 $l=1 \dots n4$

Para cada planta "j" y en cada turno "l" la decisión de producir cualquier producto "i" y en cualquier línea de producción "k" deberá considerar la cantidad de cuadrillas disponibles para producción.

Restricciones (2 de 3)

M_{ijl}		
L1	LS	
0	F_{ij}	$X_{ijkl} = 0$
F_{ij}	∞	$X_{ijkl} = 1$

$$F_{ij} * X_{ijkl} \leq M_{ijl}$$

$$F_{ijkl} + \infty X_{ijkl} \geq M_{ijl}$$

Para $i, j, k, l = 1 \dots n1, n2, n3, n4$

Para cada producto "i", en cada planta "j", en cada línea de producción "k" y en cada turno "l", se deberá de considerar la política de inventarios mínimos requeridos de insumos para poder producir.

El inventario final del producto "i" en la planta "j" y en el turno "l" será igual a: inventario final del turno anterior más el inventario actual en proceso mas la producción menos la demanda.

$$I_{ijl} = I_{ijl-1} + \sum_{k=1}^{n3} W_{ijkl} + \sum_{k=1}^{n3} R_{ik} * X_{ijkl} - D_{ijl}$$

Para $i, j, l = 1 \dots n1, n2, n4$

Restricciones (3 de 3)

$$\sum_{i=1}^{n1} I_{ijl} \leq H_j \quad \text{El inventario final en cada planta "j" y en cada turno "l" no deber\u00e1 exceder la capacidad de almacenamiento.}$$

para $j, l = 1..n2, n4$

$$I_{ijl} \geq P_{ij} \quad \text{El inventario final de cada producto "i" en cada planta "j" y en cada turno "l", deber\u00e1 considerar la pol\u00edtica de inventarios m\u00ednimos.}$$

para $i, j, l = 1..n1, n2, n4$

$$\sum_{i=1}^{n1} \sum_{h=1}^{n2} Y_{ijhl} \leq S_{jl} \quad \text{Para cada planta "j" y en cada turno "l", el total de la cantidad de producto "i" a fletear hacia todas las plantas "h", deber\u00e1 considerar la capacidad de fleteo de salida.}$$

para $j, l = 1..n2, n4$

$$\sum_{i=1}^{n1} \sum_{h=1}^{n2} Y_{ihjl} \leq E_{jl} \quad \text{Para cada planta "j" y en cada turno "l", el total de la cantidad de producto "i" recibido desde todas las plantas "h", deber\u00e1 considerar la capacidad de fleteo de entrada.}$$

para $j, l = 1..n2, n4$

Funci\u00f3n Objetivo:

$$FO_{min} : \sum_{i=1}^{n1} \sum_{j=1}^{n2} \sum_{k=1}^{n3} \sum_{l=1}^{n4} C_{ik} * R_{ik} * X_{ijkl} +$$

$$\sum_{i=1}^{n1} \sum_{j=1}^{n2} \sum_{h=1}^{n2} \sum_{l=1}^{n3} G_{jh} * Y_{ijhl}$$

Minimizar el total de los costos de:

- 1.- Producci\u00f3n.
- 2.- Fleteo.

Considerando:

- 1.- Todos los productos.
- 2.- Todas las l\u00edneas de producci\u00f3n.
- 3.- Todos los almacenes y
- 4.- Todos los turnos a querer ser programados.

La modelación anterior, aunque contiene una mayor cantidad de variables que el primer modelo expuesto en el apartado 3.2, no obstante resulta ser más fácil de resolver computacionalmente hablando. La principal razón para afirmar lo anterior tiene que ver con la dimensionalidad combinatoria de ambos problemas. Es decir, el presente problema tiene muchas menos combinaciones que atender que el problema revisado en el apartado 3.2.

La estructura matemática del presente problema parte de una situación actual determinada por las siguientes condiciones de operación:

1. Niveles de inventario por producto y por planta tanto para el producto terminado, para el producto en proceso y también para los insumos.
2. Pronósticos de demanda y políticas de inventario por producto y por planta.
3. Disponibilidad de capacidad en términos de las cuadrillas de obreros por planta y por turno.
4. Disponibilidad de capacidad en términos del programa de mantenimiento preventivo por máquina y por turno.
5. Matriz de flexibilidad productiva en términos de la capacidad de las máquinas para producir cuales productos.
6. Capacidades de almacenaje y flujo de transporte entrante y saliente en la planta.
7. Costos de producción, de almacenaje y de transporte.

La formulación central del modelo gira en torno a un problema de flujo de redes. Se parte de los inventarios iniciales en cada uno de los nodos de la red los cuales se ven modificados de acuerdo a los flujos eventuales que ocurran en la red. Así entonces se da origen al inventario final en cada nodo de la red el cual es confrontado versus las políticas que hayan sido definidas por el planeador.

Del párrafo anterior tenemos que los flujos propuestos en la red, a su vez resultan básicamente de las variables endógenas del problema. Las variables endógenas del modelo a su vez se concentran en dar una solución óptima al programa de producción de las plantas así como al programa de transporte de la red logística. Finalmente, la solución óptima del modelo se rige entonces por los criterios de costos mencionados en el inciso 7 del párrafo anterior.

3.4 Tercer problema de la logística: programación de la producción para ambientes discretos .

En los ambientes dedicados a la manufactura o al desarrollo de servicios personalizados tipo discretos (ó también denominados Job-Shop) es siempre crítico y crucial determinar en qué orden deben ser atendidas las órdenes de producción a ser procesadas en la planta. Un criterio de secuenciación normalmente utilizado para determinar el orden de ejecución de los pedidos de producción es el de "Primeras Entradas - Primeras Salidas" (FIFO). Sin embargo, este criterio de secuenciación, no obstante su simplicidad, no necesariamente resulta siempre el mejor en virtud de que las necesidades de los clientes que compiten por los recursos productivos de la empresa no tienen las mismas prioridades y exigencias.

Un ejemplo de lo anterior sería aquel caso en el cual un cliente "X" haya fincado su orden primero que otro cliente "Y". En este caso, un criterio FIFO nunca podría tomar en consideración el hecho de que al cliente "X" lo que más le importa no es un tiempo de entrega urgente sino más bien aprovechar alguna consideración o descuento de tipo económico. En este mismo caso, podría ocurrir que el cliente "Y" esté dispuesto a pagar un precio más alto por recibir un servicio express.

Obviamente, conocer las prioridades del cliente siempre será el punto de partida para cualquier negociación que promueva el eficiente uso de los recursos de la empresa, coadyuvando de esta manera a incrementar la rentabilidad del negocio. Esto no quiere decir que no se puedan ofrecer tiempos de entrega cortos al cliente, más bien la pregunta primera es, qué tanto al cliente le interesa eso como para estar dispuesto a pagar un premio. Aquí es cuando el cliente de manera racional buscará valorar ambos aspectos en la búsqueda de un justo medio.

Indistintamente, cualquier consideración de prioridades definido por parte del cliente, sería dada a conocer al modelo de secuenciación matemático con la finalidad de tomarlo en cuenta como una más de las restricciones del sistema. Lo importante es que este tipo de restricciones o eventualidades que el cliente está dispuesto a pagar por ellas sean aprovechadas de manera efectiva por las áreas involucradas en la empresa.

Ahora bien, dejando de lado los intereses y prioridades de nuestros clientes y echando una mirada al aspecto operativo y productivo, podemos luego apreciar la dificultad que conlleva el mantener una planta siempre ocupada y de hacerlo de manera equilibrada y estable evitando con esto los picos de carga de trabajo. Esto significa el tomar en cuenta factores que van desde el cumplimiento de los programas de

mantenimiento preventivos hasta la administración de los tiempos de espera y de los inventarios en proceso (WIP) que se generan a la entrada de cada una de las estaciones de trabajo por las cuales tiene que pasar un producto o servicio durante su elaboración.

Es comprensible darse cuenta que entre mayor sea la cantidad de órdenes de producción "n" que comiencen por empezar a ser elaboradas, así como también entre mayor sea la cantidad de procesos u estaciones de trabajo "m" que estén involucradas en dichas órdenes de producción, pues también crece de manera exponencial la complejidad del problema a ser resuelto. Así pues, este problema puede ser descrito en términos de su complejidad computacional como un problema del tipo "difícil de resolver" o también referidos como "NP-Hard" (del inglés Non-polinomial Hard).

La complejidad del problema se aprecia una vez que el objetivo del administrador es encontrar aquella secuencia en la cual óptimamente conviene procesar las órdenes de producción tomando en cuenta todas las restricciones presentes en el sistema. Los criterios de optimización, que a la vez pueden ser también llamadas estrategias, pueden ser tantas como las condiciones de un negocio las requiera. Aquí, por ejemplo enumeramos algunas que pudieran ser interesantes a tomar en cuenta por los administradores preocupados por resolver el problema:

1. Minimizar el total de los tiempos de retraso (horas, días, etc) del total de las órdenes de producción pendientes de ser entregadas al cliente (Backlog).
2. Minimizar el máximo tiempo de retraso de aquella orden de producción que tenga que esperar más tiempo en ser entregada.
3. Minimizar los costos financieros así como los relacionados con mermas y desperdicios ocasionados por el exceso de los inventarios en proceso (WIP).
4. Minimizar los tiempos de ocio en las estaciones de trabajo de la planta.
5. Minimizar los tiempos de espera de las ordenes de producción a la entrada de las estaciones de trabajo.
6. Maximizar la utilidad de operación.

A continuación se anexa el modelo matemático planteado para el problema en cuestión.

(M4) Secuenciación de "n" ordenes en "m" máquinas con tiempos de Setup determinísticos y tiempos de entrega negociados.

(Resource Constrained Job Shop Scheduling Problem)

Variables de Entrada:

A_i = Tiempo de Procesamiento de la Orden "i" en la Máquina "A"

B_i = Tiempo de Procesamiento de la Orden "i" en la Máquina "B"

C_i = Tiempo de Procesamiento de la Orden "i" en la Máquina "C"
.... y así para "m" máquinas que puedan estar involucradas

L_i = Tiempo de entrega negociado con el cliente de la Orden "i"

V_i = Costo de descuento para la orden "i" por cada día de retraso en su tiempo de entrega real contra el comprometido al cliente.

U_i = Precio marginal acordado con el cliente de la orden "i" por cada día de adelanto a su tiempo de entrega comprometido.

donde "i" son las ordenes de producción desde $i = 1 \dots n$

Variables de Salida (1 de 2):

$X_{ij} \geq 0, \leq 1, \text{ent} \begin{cases} 0 \rightarrow \text{La orden "i" NO será procesada en la secuencia "j"} \\ 1 \rightarrow \text{SI} \end{cases}$

$PA_j \geq 0 \begin{cases} \text{Tiempo de proceso en la máquina "A" de la orden} \\ \text{a ser procesada en la secuencia "j"} \end{cases}$
igual PB_j y PC_j para las Máquinas "B" y "C"

$TA_j \geq 0 \begin{cases} \text{Tiempo determinación en la máquina "A" de la orden} \\ \text{a ser procesada en la secuencia "j"} \end{cases}$
igual TB_j y TC_j para las Máquinas "B" y "C"

$H_j \geq 0 \begin{cases} \text{Tiempo de entrega comprometido al cliente para} \\ \text{la orden procesada en la secuencia "j"} \end{cases}$

Variables de Salida (2 de 2):

$$R_j \geq 0 \left\{ \begin{array}{l} \text{Tiempo de retraso respecto a la fecha de entrega comprometida} \\ \text{con el cliente de la orden a ser procesada en la secuencia " j" } \end{array} \right.$$

$$S_j \geq 0 \left\{ \begin{array}{l} \text{Tiempo de adelanto respecto a la fecha de entrega comprometida} \\ \text{con el cliente de la orden a ser procesada en la secuencia " j" } \end{array} \right.$$

$$W_j \geq 0, \leq 1, \text{ent} \left\{ \begin{array}{l} 0 \rightarrow \text{La orden a ser procesada en la secuencia " j" } \\ \text{NO tuvo retraso} \\ 1 \rightarrow \text{SI} \end{array} \right.$$

$$Z \geq 0 \left\{ \text{Máximo tiempo de retraso obtenido para las ordenes en el sistema} \right.$$

$$F_j \geq 0 \left\{ \begin{array}{l} \text{Costo de descuento diario comprometido con el cliente por incumplimiento} \\ \text{en la fecha de entrega para la orden procesada en la secuencia " j" } \end{array} \right.$$

$$G_j \geq 0 \left\{ \begin{array}{l} \text{Sobre precio marginal diario negociado con el cliente por adelanto} \\ \text{en la fecha de entrega para la orden procesada en la secuencia " j" } \end{array} \right.$$

Restricciones (1 de 2):

Restricciones para normalización de las variables inputables a :

a) Tiempos de proceso para la orden " i" en cada máquina (A, B y C)

b) Tiempo de entrega comprometido con el cliente para cada orden " i"

c) Descuentos y Precios marginales negociados con el cliente por retraso- adelanto

$$PA_j = \sum_{i=1}^n X_{ij} * A_i \quad PB_j = \sum_{i=1}^n X_{ij} * B_i \quad PC_j = \sum_{i=1}^n X_{ij} * C_i \quad H_j = \sum_{i=1}^n X_{ij} * L_i$$

$$F_j = \sum_{i=1}^n X_{ij} * V_i \quad G_j = \sum_{i=1}^n X_{ij} * U_i$$

Cálculo del tiempo de terminación de la orden " j" en la máquina " A" ; para j = 1 ... m $\therefore TA_j = TA_{j-1} + PA_j$

Cálculo del tiempo de ocio y del tiempo de espera para la orden "j" en la entrada de la máquina "B" y "C" ; para j = 1 ... m

$$TA_j - TB_{j-1} = OB_j - EB_j$$

$$TB_j - TC_{j-1} = OC_j - EC_j$$

Restricciones (2 de 2):

Cálculo del tiempo de terminación para cada orden "j" en las máquinas "B" y "C"; para $j = 1 \dots m$

$$TB_j = TA_j + EB_j + PB_j \qquad TC_j = TB_j + EC_j + PC_j$$

Cálculo del tiempo de retraso-adelanto en el sistema; para $j = 1 \dots m$

$$TC_j - H_j = R_j - S_j$$

Cálculo del máximo retraso en el sistema; para $j = 1 \dots m$ $\therefore Z \geq R_j$

Cálculo de la cantidad de tareas con retraso en el sistema; para $j = 1 \dots m$

R_j			
$\frac{LI}{LS}$			
0	0	$W_j = 0$	$W_j \leq R_j$
1	∞	$W_j = 1$	$\infty W_j \geq R_j$

Funciones Objetivo:

$$FO_{min} : \sum_{j=1}^m W_j \quad \left\{ \text{Minimizar la cantidad de ordenes con retraso} \right\}$$

$$FO_{min} : \sum_{j=1}^m R_j \quad \left\{ \begin{array}{l} \text{Minimizar la cantidad de dias totales} \\ \text{con retraso en el sistema} \end{array} \right\}$$

$$FO_{min} : Z \quad \left\{ \begin{array}{l} \text{Minimizar el máximo retraso de alguna} \\ \text{orden en el sistema} \end{array} \right\}$$

$$FO_{min} : TC_n \quad \left\{ \begin{array}{l} \text{Minimizar el tiempo de ocio en el sistema; osea minimizar} \\ \text{el tiempo de terminación de la última orden en el sistema} \end{array} \right\}$$

$$FO_{min} : \sum_{j=1}^m (R_j + S_j) \quad \left\{ \begin{array}{l} \text{Minimizar la cantidad de dias de retraso y} \\ \text{adelanto en el sistema, "Just-in-Time"} \end{array} \right\}$$

$$FO_{max} : \sum_{j=1}^m (S_j * G_j - R_j * F_j) \quad \left\{ \begin{array}{l} \text{Maximizar la ganancia global :} \\ \text{pedidos con adelanto versus} \\ \text{pedidos con retraso} \end{array} \right\}$$

Recordemos que el primero modelo presentado en el apartado 3.2 correspondió a la clasificación de problemas de “empacamiento” (o Packing en inglés). El segundo en el apartado 3.3 correspondió a uno de “flujo de redes” (o Network Flow en inglés). El presente modelo expuesto en el apartado 3.4 corresponde a la clasificación de problemas del tipo de “secuenciamiento” (o Scheduling en inglés). En general, los problemas de secuenciamiento así como los de empacamiento son problemas NP-Hard (en el capítulo 5 se tratará este tema).

Básicamente la estructura matemática del problema gira en torno al conjunto de variables que identifican la cantidad de tiempo de preparación (o setup time en inglés), así como del tiempo productivo requerido para procesar cada orden en cada una de las máquinas por donde deba ser procesada. Adicionalmente el modelo incorpora una variable de tiempo y otras dos variables económicas para definir la importancia relativa que el planeador está tomando en cuenta para decidir en que momento conviene entregar la orden al cliente.

El esfuerzo del modelo matemático se enfoca en el cálculo del tiempo de finalización de cada orden. Naturalmente dicho tiempo ocurre una vez que concluye su proceso en la última máquina por donde debe pasar. Por tal motivo, es razonable observar que en la modelación se desarrolla un conjunto de restricciones con la finalidad de dar consistencia al momento en el que cada orden entra a procesarse en cada máquina. Dicho momento depende lógicamente del momento en que cada orden haya finalizado su procesamiento en la máquina anterior así como del tiempo en el cual haya finalizado su procesamiento la orden que aparece en una secuencia adelante en la máquina a la cual se requiere entrar. Con lo anterior, es entonces posible determinar la estrategia óptima en la cual conviene que cada orden se procese en las máquinas productivas. Lo anterior consecuentemente determina el tiempo de terminación de cada orden y por ende un tiempo de retraso o de adelanto con respecto a la fecha de entrega comprometida al cliente.

Aunque el problema de secuenciamiento tratado en el presente apartado está enfocado a dar tratamiento a problemas relacionados en el ámbito de la manufactura, no obstante posee ciertas similitudes matemáticas respecto al cuarto problema general de la logística que será expuesto en el siguiente apartado 3.5 el cual se concentra, a diferencia del anterior, en problemas de ruteo de distribución. La razón de la similitud anterior obedece al hecho de que ambos problemas caen matemáticamente dentro de la categoría de problemas de “secuenciamiento”. No obstante lo anterior, a los problemas de “secuenciamiento” cuya aplicación es en ruteo de distribución se les categoriza simplemente como problemas de ruteo (o Routing en inglés).

3.5 Cuarto problema de la logística: programación de la distribución (Ruteo).

Uno de los problemas más difíciles a resolver en la teoría matemática contemporánea corresponde a determinar aquella secuencia óptima en la cual llevar a cabo un programa de eventos o actividades en los cuales no existe una precedencia determinada. El objetivo en la operación logística de distribución es la obtención de un programa o secuenciación de entrega que asegure hacer llegar el producto en la cantidad adecuada, en el momento acordado con el cliente y de la manera menos costosa posible.

El objetivo de una programación óptima de la distribución no es otra cosa que "operar" de la manera más eficiente posible la distribución física; apoyándose en el plan estratégico de distribución logístico ya antes cubierto. En términos específicos, nos referimos a obtener el recorrido y la secuencia óptima de los clientes a ser visitados a través de la configuración de una ruta de distribución. A este problema en su variante más básica se le conoce como el "Problema del Agente Viajero" o "TSP" (Traveling Salesman Problem).

No obstante, podemos generalizar este fenómeno hacia una configuración en la cual se tengan que determinar no solamente la secuenciación de los clientes a ser atendidos por una ruta, sino también la asignación de cada uno de los clientes a cada una de las diversas rutas o medios de distribución que se tengan disponibles de tal manera que también sea necesario optimizar la cantidad de vehículos a ser utilizados. A este último problema que resulta aún más general se le denomina "Problema de Ruteo de Vehículos" o "VRP" (Vehicle Routing Problem).

Así pues, un Programa de Distribución óptimo debería tomar en consideración:

1. Ubicación geográfica de los clientes.
2. Restricciones de orientación de calles.
3. Normas y regulación de vialidad y tránsito (horarios de circulación, límites de velocidad, vueltas a la derecha o a la izquierda o en "u", etc).
4. Restricciones de horario que puedan haber por parte del cliente para su atención.
5. Usos y costumbres de entrega de acuerdo a las barreras topográficas que pudieran presentarse.
6. Consideraciones de "paradas únicas" para entrega o recolección de producto correspondiente a clientes múltiples, modalidad para entrega en centros comerciales, etc.

Así pues, habiendo tomado en cuenta todas las consideraciones antes mencionadas, podemos definir conceptualmente el problema, como aquél en el cual se busca encontrar aquella secuencia o recorrido en el cual visitar a los clientes a ser atendidos por una ruta o varias rutas a la vez de tal manera que se optimicen cualquiera de los siguientes criterios que el administrador decida convenientemente aplicar en caso particular:

1. Minimizar las distancias recorridas.
2. Minimizar los costos incurridos.
3. Minimizar los tiempos de retraso.
4. Minimizar # de vehículos requeridos.

No obstante estos criterios de optimización pudieran parecer entre ellos mismos relaciones lineales, cada uno de ellos debiera tratarse por separado de acuerdo a las necesidades y criterios que el administrador mejor considere. El modelo matemático para este problema será presentado en extenso en el capítulo 6 ya que éste es precisamente el problema de investigación del proyecto doctoral.

Concluida la exposición tanto conceptual como del modelo matemático para los cuatro problemas generales en la logística, pasaremos en el siguiente apartado 3.6 a revisar cuál es la relación que guardan los sistemas ERP y APS con respecto al OR/MS en lo general y con respecto a la contribución que este tipo de sistemas han brindado para solucionar los problemas generales de la logística en lo particular. Derivado de la exposición teórica bibliográfica que se hará en el siguiente apartado, se procederá a continuación en el apartado 3.7 a realizar un análisis comparativo específico a la funcionalidad que las diversas ofertas de este tipo de sistemas comercialmente ofrecen para solucionar nuestro problema de investigación (problemas de ruteo de distribución).

Finalmente cerramos el capítulo con el apartado 3.8 en el cual se expone un estudio de caso. Dicho estudio de caso tiene el objetivo de determinar la incidencia actual de las empresas grandes y medianas del área metropolitana de la ciudad de Monterrey en cuanto a las metodologías que éstas aplican para resolver los problemas relacionados en el ámbito logístico en general. Mediante dicho estudio de caso, pretendemos contrastar los resultados obtenidos respecto a la teoría expuesta en el apartado 3.6.

3.6 La relación de los sistemas MRP, ERP y APS con el OR/MS.

A fin de lograr una mejor comprensión del tema, invitamos al lector a revisar el significado de las siglas MRP, ERP y APS en el glosario de la tesis. Un problema que enfrenta la profesión del OR/MS es el acceso a la información relevante. Esto es crucial, sobre todo tomando en cuenta que una de las cualidades del OR/MS es la de generar valor a partir de la información. En los sistemas de información de las empresas, los datos son guardados obedeciendo a políticas de acceso y temporalidad. En otros casos se observa tan solo, que estos datos son mantenidos de manera informal, irregular o de manera inconsistente.

La dificultad de los proyectos de OR/MS y en general de la mayor parte de aquellos que buscan resolver un problema, tiene que ver con la dificultad humana para distinguir, ordenar y hacer uso de la información que es relevante. Lo anterior toma relevancia asumiendo que la información relevante proviene de un océano de datos donde también está mezclada con otros datos que no son relevantes. No es aventurado asegurar que los proyectos de OR/MS evidencien la falta de calidad y de relevancia de los datos que puedan estarse almacenando en los sistemas de información de las empresas. Los Sistemas ERP han contribuido a la disponibilidad de información en las organizaciones. Sin embargo, no existe la misma facilidad para hacer uso de éstos en la toma de decisiones de la logística. Ambos requerimientos (uno cubierto y el otro no) son complementarios. En la Figura 3.2 se evidencia la relación entre ambos.

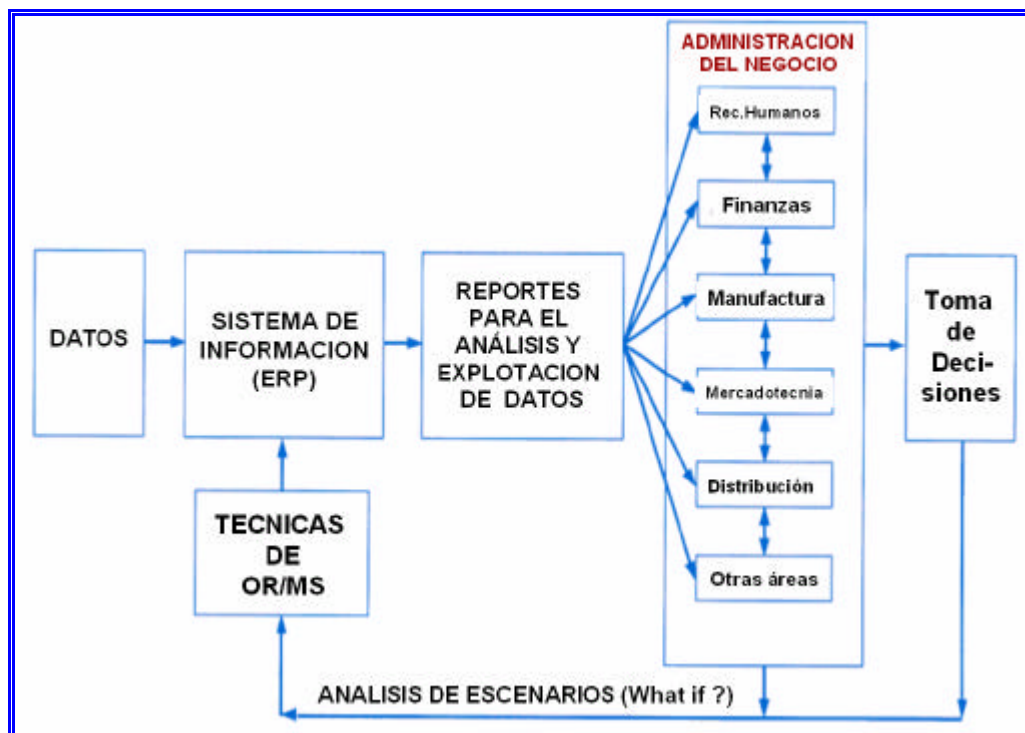


Figura 3.2 Relación de los sistemas ERP con el OR/MS en el proceso de toma de decisiones.

De los problemas generales de la logística presentados anteriormente, verificamos que tres de ellos tienen algo en común. Es decir, (1) programación de la producción para ambientes continuos, (2) programación de la producción para ambientes discretos y (3) programación de la distribución o ruteo; poseen un componente común llamado "secuenciación". La programación o secuenciación (en inglés Scheduling), consiste en encontrar la mejor secuencia de tiempo de inicio para cada actividad involucrada en el proceso en cuestión, de tal manera que logre el menor costo de operación para el sistema y al mismo tiempo la mejor oferta de servicio al cliente.³¹

El Scheduling involucra ordenar los eventos dependientes unos de otros y en secuencia; considerando la demanda, tiempos de entrega, inventarios y restricciones de capacidad que son utilizados para determinar la mejor asignación de recursos y la mejor secuenciación de actividades para lograr un equilibrio entre los costos de la logística y el nivel de servicio al cliente. Otra forma de definir el significado de "Scheduling" es en términos de las soluciones que ofrece. Así pues podemos entonces decir que "Scheduling" consiste en definir:

1. ¿Qué debo producir o distribuir?
2. ¿Cuándo debo hacerlo?
3. ¿En qué cantidad debo hacerlo (Cuánto)?
4. ¿Desde dónde debo hacerlo?
5. ¿Hacia dónde debo hacerlo?
6. ¿Cómo debo hacerlo (en que secuencia)?
7. ¿Que recursos requiero para hacerlo (con qué)?

Las interrogantes anteriores se deben resolver sujetas a restricciones de demanda, capacidades de recursos (máquinas, gente, dinero, etc), tiempos de entrega, inventarios, etc. y buscando a la vez alcanzar objetivos tales como:

1. Maximizar la rentabilidad de la empresa.
2. Minimizar el tiempo de entrega al cliente / Maximizar el nivel de servicio
3. Minimizar los tiempos de ocio / Minimizar los tiempos de espera.

³¹ Hartmann, Sonke. (2000), Project Scheduling Under Limited Resources: Models, Methods, and Applications, Springer Verlag, EUA, pag 35-60.

La programación o secuenciación es un problema comúnmente enfrentado en las empresas de manufactura y de servicios que habitualmente se administra de manera manual y sobre la base de la experiencia de los administradores. El programador es el responsable de minimizar las ineficiencias (costos) en el sistema y de maximizar el nivel de servicio al cliente. Encontrar la solución óptima a los problemas de "Scheduling" representa un desafío en el ámbito de las teorías matemáticas contemporáneas, en el siguiente capítulo 4 será tratado este tema de manera formal.

Más del 20% de las grandes empresas en el mundo han gastado miles de millones de dólares en resolver sus problemas de logística. Ello ha evolucionado gradualmente iniciando desde los 70's con los Sistemas MRP hasta llegar a los 90's con las versiones de sistemas ERP. En la actualidad, los proveedores de sistemas ERP ya aceptaron que sus supuestas soluciones ofrecidas en el pasado no resuelven este tipo de problemas referentes a la logística. Dado que los sistemas MRP y ERP no poseen la inteligencia necesaria, las corporaciones han tenido que resolver con desarrollos propios el problema ³².

El Phd. Thomas Hsiang, director en la Corporación Sensient Technologies con más de 40 plantas alrededor de 20 países en el mundo con sede en Milwaukee. Junto con Hsiang, tenemos a John Liu y D. Q. Yao ambos son profesores de "Management Science" de la Escuela de Negocios en la Universidad de Wisconsin en Milwaukee. Ellos tres, en su publicación "La Ilusión del Poder" detallan sus puntos de vista respecto a investigaciones que han hecho con empresas en los Estados Unidos en donde han tenido que resolver problemas relacionados a la Logística en general, a continuación cito (traducción libre):

"No importa qué tanto los proveedores de sistemas MRP y ERP digan que sus sistemas pueden optimizar la producción y los inventarios; sin importar que sus vendedores argumenten las capacidades de sus sistemas de información— duden de ellos!. No he encontrado un solo sistema que verdaderamente optimice. Todos los sistemas actualmente en el mercado son básicamente simplistas hasta el punto de caer en la ingenuidad. La experiencia obtenida en mis años de consultoría me hacen concluir lo siguiente:

- 4. Los profesionales del ORMS pueden contribuir grandemente a mitigar la falta de funcionalidad inherentemente encontrada en los actuales sistemas MRP/ERP*
- 5. Mientras tanto las propuestas de solución arrojados por los sistemas MRP/ERP deben ser revisadas con mucha reserva."*

³² Hsiang, Thomas. (2001), The Illusion of Power, "ORMS Today", EUA, pag 5.

Los proveedores de sistemas ERP de los 90's, son los mismos que ahora están ofreciendo la promesa de sus sistemas APS (Advanced Planning Scheduling) como remedio final a los problemas de planeación y secuenciación de la logística. Sin sistemas de planeación y de secuenciación, aún los mejores sistemas de operación ofrecerán soluciones que costarán más de lo necesario y sin alcanzar los niveles de servicios deseados por el cliente. Esta fue la triste experiencia de las empresas que implementaron sistemas ERP y APS los cuales, aunque buscaban planes y programas óptimos, lo que la mayor parte de los "software" disponibles ofrecieron fueron soluciones que quizás eran aprovechables pero bastante lejos de lo óptimo.³³

Aunque por un lado los sistemas ERP ofrecieron una mejora en la cantidad y en la calidad de los datos que el OR/MS requería, por el otro lado, algunas compañías decidieron dejar en desuso los modelos de OR/MS ya antes desarrollados con la errónea creencia de que estos sistemas ERP podrían sustituir todas las decisiones de planeación y secuenciación requeridas en la función logística. Pero los profesionales del OR/MS dejaron que lo anterior ocurriera y entonces los proveedores de sistemas APS empezaron a inundar el mercado con software y heurísticas simplistas para atender las necesidades logísticas de las empresas.³⁴ Entre estos proveedores tenemos incluidos a Aspen Tech, D Technologies, Manugistics, Numetrix y PeopleSoft. Ellos convencieron a los proveedores de sistemas ERP a reconocer la necesidad de ahora sí ya cubrir la funcionalidad requerida en la planeación y secuenciación logística que viniera a complementar sus sistemas.

Se pudiera argumentar que al final, las herramientas del OR/MS están ya embebidas en los sistemas APS, sin embargo, el problema no es ese. Mientras las empresas no asocien el uso de estos sistemas con las "competencias" requeridas por el OR/MS, habrá un pobre aprovechamiento de las inversiones que se vienen haciendo en estos sistemas. Como veremos a continuación en el estudio de campo, se verificará que en la generalidad de las empresas de la ciudad de Monterrey que han implementado algún tipo de sistema APS; aún no se han alcanzado los resultados esperados. Estas "competencias" antes mencionadas en el profesional del OR/MS, ya sean aplicadas a través de un sistema APS o a través del desarrollo de un modelo propio, propician que nuestro proyecto de investigación doctoral se enfoque en ofrecer una solución especializada para un problema particular de ruteo de distribución, el cual se expondrá con todas sus propiedades en el capítulo 6. Así entonces a través del racionamiento anterior, podemos decir que se origina nuestro proyecto de investigación de tesis.

³³ Geoffrion, Arthur. (2000), The value of operations research, "ORMS Today", EJA, pag 2.

³⁴ Sodhi, ManMohan. (2001), A Match Made in Heaven, "ORMS Today", EJA, pag 1.

3.7 Identificación y análisis comparativo respecto a las alternativas de software comercial tipo APS disponibles en el mercado para el tratamiento de problemas de distribución de ruteo.

La elaboración del presente análisis se desarrolló durante el mes de Febrero del 2003 sobre la base de información del dominio público propiedad de cada una de estas empresas de software vía folletos comerciales e intensivamente a través de recopilación de información disponible vía electrónica en internet. El presente análisis comparativo se desarrolla con la finalidad de atender 5 objetivos primordiales:

1. Identificar cual es la oferta de software comercial APS actualmente disponible en el mercado.
2. Determinar que tanto arraigo (antiguo) tiene el software en el mercado.
3. Diferenciar los elementos de capacidad de cada una de éstas ofertas.
4. Identificar el tipo de industria hacia el cual preponderantemente está orientado el software.
5. Medir la cantidad de empresas que actualmente lo utilizan.

En la primera página aparecen las Tablas 3.1 (a) y (b). Ambas tablas muestran la misma información por renglón, simplemente lo que complementa una a la otra son las empresas que aparecen en cada una de las columnas. Como puede apreciarse, estas tablas buscan atender los primeros 3 objetivos que planteamos al inicio de este apartado.

En la segunda página aparecen las Tablas 3.2 (a) y (b). Aquí aplica el mismo comentario hecho en el párrafo anterior. Es decir estamos hablando de tablas complementarias. Como se va a poder más adelante constatar, estas tablas buscan atender los últimos dos objetivos planteados al inicio del apartado.

Tablas 3.1 (a) y (b) Comparativa de software para optimización para aplicaciones de ruteo.

		1	2	3	4	5	6	7
SOFTWARE		Direct Route	ILOG Dispatcher	Intertour / Interload	Manugistics Fleet Management	Protour	RoadNet 5000	Territory Planner
PROVEEDOR		Appian Logistics Software	ILOG	Intertour-Interload	Manugistics	Prologos Planung und Beratung	UPS Logistics Group	UPS Logistics Group
AÑO DE INTRODUCCIÓN		1996	1997	1983	1985	1994	1983	1983
DIMENSION MAXIMA DEL PROBLEMA	# DE NODOS	16,000	Sin Límite	850/8,000	Sin Límite	4,000	Sin Límite	Sin Límite
	# DE RECURSOS	1,000	Sin Límite	500/8,000	Sin Límite	1,000	Sin Límite	Sin Límite
ALGORITMO GENERAL DE SOLUCION	Secuenciador de Ruteo	Y	Y	Y	Y	Y	Y	Y
	Asignación de Nodos por Ruta	Y	Y	Y	Y	Y	Y	Y
	Modelación basada en Distancias Geodésicas	Y	Y	Y	Y	Y	Y	Y
	Ruteo de tráfico en Tiempo Real	-	-	Y	Y	-	-	-
	Modelación Dinámica de Atributos Relacionados al Arco (Flujo y Velocidad)	-	-	Y	Y	Y	Y	Y
	Modelación de Atributos estocásticos (Demandas y Tiempos)	Y	-	Y	Y	Y	Y	Y
	Ventana de Horario Flexible (Con Penalización)	Y	Y	Y	Y	Y	Y	Y
	Ventana de Horario Rígida	-	Y	Y	Y	Y	-	-
	Asignación de Recursos para Ruteo (Vehículos, Operadores)	Y	Y	Y	Y	-	Y	Y
	Modelación de Atributos de Vialidad y Tránsito	Y	-	Y	Y	Y	Y	Y
	Modelación de Restricciones y Capacidades de Carga y Descarga	Y	-	Y	Y	-	-	-

		8	9	10	11	12	13	14	15
SOFTWARE		Roadshow System	RoutePro	Route Smart	SHORTREC product suite	STARS (Smart Truck Assignment and Routing System)	tmsRouter	Trapeze	TruckSTOPS Routing & Scheduling for Windows
PROVEEDOR		Descartes Systems Group	CAPS Logistics	RouteSmart Technologies, Inc.	ORTEC	SAITECH, Inc.	GeoComtms	Trapeze Software Group, Inc.	Micro Analytics, Inc.
AÑO DE INTRODUCCIÓN		1987	1997	1989	1981	1995	2000	1992	1984
DIMENSION MAXIMA DEL PROBLEMA	# DE NODOS	32,000+	-	Sin Límite	Sin Límite	100	Sin Límite	Sin Límite	Sin Límite
	# DE RECURSOS	Sin Límite	-	Sin Límite	Sin Límite	1,000	Sin Límite	Sin Límite	Sin Límite
ALGORITMO GENERAL DE SOLUCION	Secuenciador de Ruteo	Y	Y	Y	Y	Y	Y	Y	Y
	Asignación de Nodos por Ruta	Y	Y	Y	Y	Y	Y	-	Y
	Modelación basada en Distancias Geodésicas	Y	Y	Y	Y	Y	Y	Y	Y
	Ruteo de tráfico en Tiempo Real	Y	-	-	-	-	Y	Y	-
	Modelación Dinámica de Atributos Relacionados al Arco (Flujo y Velocidad)	Y	Y	Y	Y	Y	Y	Y	-
	Modelación de Atributos estocásticos (Demandas y Tiempos)	Y	Y	Y	Y	-	-	Y	Y
	Ventana de Horario Flexible (Con Penalización)	Y	Y	Y	Y	Y	Y	Y	Y
	Ventana de Horario Rígida	Y	Y	-	Y	Y	Y	Y	-
	Asignación de Recursos para Ruteo (Vehículos, Operadores)	Y	Y	-	Y	Y	Y	Y	Y
	Modelación de Atributos de Vialidad y Tránsito	Y	Y	-	Y	Y	Y	Y	Y
	Modelación de Restricciones y Capacidades de Carga y Descarga	Y	-	-	Y	Y	Y	-	Y

Tablas 3.2 (a) y (b) Comparativa de software de optimización para aplicaciones de ruteo (2da Parte).

SOFTWARE		1	2	3	4	5	6	7
SOFTWARE		Direct Route	ILOG Dispatcher	Intertour / Interload	Manugistics Fleet Management	Protour	RoadNet 5000	Territory Planner
TIPO DE CLIENTES ACTUALMENTE UTILIZAN EL SOFTWARE	Transporte de Carga Pesada	Y	Y	Y	Y	Y	Y	Y
	Entrega y Recolección en Ruta Local	Y	Y	Y	Y	Y	Y	Y
	Paquetería y Mensajería	Y	Y	Y	-	Y	-	-
	Transporte de Pasajeros	-	Y	Y	-	-	-	-
	Taxis	-	Y	Y	-	-	-	-
	Servicio de Entrega a Domicilio	Y	Y	Y	-	Y	-	-
CARACTERISTICAS Y COMENTARIOS		Algoritmo para Secuenciamiento Optimo de Recursos	Algoritmos para Optimización de Recolección y Entrega así como de Cubicación de Carga.	Optimización de Carga y Planeación de Transporte Multimodal	Algoritmos de Ruteo con restricciones de Tránsito y Vialidad. Manejo de Entregas Múltiples en cada Nodo.	Optimización y Balanceo de Territorios	Algoritmos de Ruteo con restricciones de Tránsito y Vialidad. Manejo de Entregas Múltiples en cada Nodo.	Optimización y Balanceo de Territorios
# DE EMPRESAS UTILIZANDO EL SOFTWARE		175	300+	500	1000+	20	1000+	1000+
PRINCIPALES CLIENTES		Walgreens, Scheider, Nabisco	-	Swiss Post, Unilever, Daimler Chrysler, Volkswagen, BMW, Rockwool, Auchan, Feldschloessen	Advance Auto, Ahold, Albertsons, Baxter Healthcare, Fleming, Kroger, Nash Finch, Perseco, Winn Dixie, Safeway, McLane	Servicio de Paquetería en Alemania	Pepsi, Sysco, U.S. Foods	Pepsi, Sysco, Anhesuser-Busch

SOFTWARE		8	9	10	11	12	13	14	15
SOFTWARE		Roadshow System	RoutePro	Route Smart	SHORTREC product suite	STARS (Smart Truck Assignment and Routing System)	tmsRouter	Trapeze	TruckSTOPS Routing & Scheduling for Windows
TIPO DE CLIENTES ACTUALMENTE UTILIZAN EL SOFTWARE	Transporte de Carga Pesada	Y	Y	-	Y	Y	Y	-	Y
	Entrega y Recolección en Ruta Local	Y	Y	Y	Y	Y	Y	-	Y
	Paquetería y Mensajería	Y	-	-	Y	Y	Y	-	Y
	Transporte de Pasajeros	-	-	-	Y	-	-	Y	-
	Taxis	-	-	-	Y	-	-	-	-
	Servicio de Entrega a Domicilio	Y	-	Y	Y	Y	Y	-	Y
CARACTERISTICAS Y COMENTARIOS		Planeación de Ruteo y Algoritmos de Despacho Dinámico.	Optimización y Balanceo de Territorios	Orientado al ámbito Postal, Recolección de Basura y Entrega de Periódico.	Optimización y Balanceo de Territorios	Algoritmos para Secuenciamiento y Asignación de Cargas de Fleteo entre Almacenes Múltiples	Basado en Algoritmos Genéticos.	Secuenciamiento para Ruteo Fijo.	Algoritmos para estrategias de Entrega y Recolección con restricciones de Capacidad de Carga y con Salidas Múltiples
# DE EMPRESAS UTILIZANDO EL SOFTWARE		1500	1000+	100+	100+	20	NO DISPONIBLE	800	2400
PRINCIPALES CLIENTES		Coca-Cola, Ambev, FedEx, Air Ground Freight Service Division	Daimler Chrysler, Ford Motor, Excel Logistics, GA-Pacific, Fresh Express	FedEx home Delivery, USPS, New York Times, Coned, 50+ municipalidades	BP, Texaco, TNT Logistics, Lafarge, Yellow Freight System	Compañía Petrolera Nisseki-Mitsubishi Oil (Japón)	Carroll Independent Fuel, Southern Maryland Oil, Purolator, Multi-Marques	Depto de Tránsito de las Ciudades de Dallas, Austin y Tucson	Home Depot, Simmons Mattress, UPS Worldwide, 7-Eleven, Dominoes, Nestle, Pepsi, Miller Brewing, Chicago Tribune

Después de haber revisado la información comparativa del software comercial disponible, lo que se desarrollo a continuación fue una investigación más específica respecto a cuáles de estos productos tienen una presencia relevante en las empresas en México. Lo anterior simplemente se hizo sobre la base de la consulta de la cartera de clientes de estas compañías que se identificaron como las de mayor presencia en México. A continuación en la siguiente hoja, aparecen las Tablas 3.3 (a) y (b) en las cuales se adiciona información comparativa sobre la base de las limitantes técnicas de cada uno de estos software's que fueron identificadas por el postulante en función a los requerimientos que se buscan solucionar en los problemas generales de la logística de ruteo.

Es importante precisar que con el presente análisis nunca se buscó concluir con algún juicio de valoración o recomendación respecto a cual de estas ofertas de software resultara mejor. En resumen, el provecho que se obtiene de este análisis comparativo, es el de identificar las diferencias funcionales que tienen cada uno de estas ofertas de software y así evidenciar lo improbable que resulta el que una sola alternativa de software APS "genérico" logre incluir todos los aspectos particulares que requieren ser considerados para los diversos problemas de logística.

Así entonces se fundamenta la necesidad de generar una propuesta de solución especializada para nuestro problema particular de ruteo de distribución el cual se planteará de manera formal en el capítulo 6. De nuevo, es precisamente la necesidad de desarrollar una propuesta de solución especializada lo que sustenta el proyecto de investigación.

Tablas 3.3 (a) y (b) Comparativa del software para planificación de rutas.

Nombre del Sistema APS	Road Show	Route Pro	Road Net	Solución Ideal
Nombre del Proveedor	Descartes	Caps Logistics	UPS Logistics	
Oficinas Corporativas	Atlanta Georgia	Atlanta Georgia	Baltimore Maryland	
Costo x Licencia	\$ 60 mil US Dlls + Consultoría	\$ 50 mil US Dlls Condicionado a 2 licencias mínimo	\$ 35 mil US Dlls y condicionado a \$30 mil US dlls en gastos de Consultoría	
# de Sesiones celebradas por Requerimiento de evaluación para Coca Cola Procor	2	4	4	
Personal Involucrado de la Compañía	Solo Gerentes de Ventas	Gerentes de Ventas y Gerentes de Sistemas	Gerentes de Ventas y Gerentes de Sistemas	
Fechas de Evaluación	Primeras 2 semanas de Marzo	Ultimas 2 semanas de Abril y primeras 2 de Mayo	Ultimas 2 semanas de Marzo y primeras 2 de Abril	
Lugares de Evaluación	Saltillo, México y Monterrey	México y Monterrey	Culiacan, Chihuahua y Monterrey	
Implementaciones en México	Coca Cola Panamco, Coca Cola Arma	Sabritas, Maseca, Gamesa, Coca Cola Femsa, Pepsi Gemex	Coca Cola Argos	
Nivel de Servicio y Soporte Técnico	Muy limitado y solo desde las oficinas en Atlanta USA	Aceptable y con oficinas solo en la Cd. De México, Joint Venture con GEDAS y BAAN	Oficina de Ventas en Monterrey, pero soporte técnico solo desde Baltimore USA	
Asignación de Territorios a Cedis	NO	Optimización: basada en cargas de trabajo	Heurístico: basado en Centros de Gravedad Geográficos	Optimización: basada en criterios Multiobjetivos
Efectividad Matemática de la Solución	NO	SI	SI	

Nombre del Sistema APS	Road Show	Route Pro	Road Net	Solución Ideal
Efectividad Práctica de la Solución	NO Maneja restricciones de Asignación o Des-asignación Dura	NO Maneja restricciones de Asignación o Des-asignación Dura	NO Maneja restricciones de Asignación o Des-asignación Dura	Se requiere capacidad para la inclusión de consideraciones de aspecto práctico
Balanceo de Cargas de Trabajo entre los Territorios	Heurístico: basado en el método de residuos	Heurístico: basado en el método de residuos	Heurístico: basado en el método de asignación secuencial	Optimización: basada en criterios Multiobjetivos
Cantidad de Objetivos o Criterios de Balanceo	Limitado a 3 de Tipo Secuencial Heurístico	Limitado a 3 propietarios, 30 definidos por el usuario de Tipo Paralelo pero Heurístico	Limitado a 3 propietarios, 10 definidos por el usuario pero todos de tipo Secuencial Heurístico	Multiobjetivo Paralelo basado en Optimización Matemática
Efectividad Matemática del Balanceo	NO	NO	SI	
Manejo de Restricciones Topográficas	NO	Utiliza Restricciones Suaves	Utiliza Restricciones Suaves	Manejo de Restricciones Duras
Manejo de Restricciones Duras de Agrupamiento o NO Agrupamiento	NO	NO	NO	Se requiere para la inclusión de restricciones de índole práctico
Manejo de Restricciones de Vialidad	Velocidades y Sentidos	Velocidades y Sentidos	Velocidades y Sentidos	Velocidades y Sentidos
Método de Clusterización y delimitación Geográfica	Arreglos de Polígonos Irregulares	Agrupamiento Irregular de Manzanas	Arreglos de Polígonos Irregulares	Agrupamiento Regular de Manzanas
Efectividad Práctica de la Solución	NO	NO	NO	
Principal Atributo Técnico a favor	Pionero en la oferta de soluciones a la Industria	Uso de Algoritmos para el Agrupamiento de Manzanas	Utilización de Heurísticas efectivas y veloces	
Principal Atributo Técnico en contra	Poco estructurado y con captura de información y de parámetros muy intensiva	Muy evidente falta de optimización matemática. Resultados pobres	No es posible el manejo de escenarios multiobjetivos y además el agrupamiento es basado en polígonos irregulares	
Calificación Final de Efectividad	20%	40%	60%	

3.8 Caso de estudio: revisión de las estrategias de solución implementadas en la ciudad de Monterrey para el apoyo en la toma de decisiones de la función logística.

En abril y mayo del año 2002, se desarrolló un experimento de campo para determinar la incidencia actual de las empresas grandes y medianas del AMM en cuanto a las metodologías que éstas aplican para resolver los problemas relacionados en el ámbito Logístico. Este tipo de empresas resultan lo suficientemente pertinentes como para requerir el empleo de herramientas de análisis cuantitativo. Además, de acuerdo a la empírica del postulante, coincide en que estas son el tipo de empresas o que ya invirtieron en Sistemas ERP-APS's o que al menos están por hacerlo. A continuación se muestra una tabla para la identificación y delimitación de la población de estudio.

Tabla 3.4 Distribución de empresas del AMM de acuerdo al sector industrial³⁵.

Identificación y delimitación de la población de estudio					
# DE EMPRESAS	TIPO EMP.				
Sector	GRANDE	MEDIANA	INTERMEDIA	MICRO-PEQUEÑA	Total general
INDUSTRIAL	60	47	72	109	288
SERVICIOS	12	15	21	51	99
VARIOS	11	10	12	52	85
CONSTRUCCION	8	12	14	36	70
ALIMENTICIO	22	14	15	14	65
COMERCIO	12	6	5	31	54
AUTOMOTRIZ	11	8	10	15	44
FINANCIERO	5	2	2	10	19
TEXTIL	6	1	2	4	13
EDUCACION	3	1	2	3	9
FARMACEUTICO	1	1	2	5	9
EDITORIAL	1	1	1	2	5
Total general	152	118	158	332	760
Población de empresas del proyecto de investigación =					235

De acuerdo al censo del GUIA-ROJI del 2000 (las 760 empresas de la guía-roji pueden variar con otros censos), tenemos un total de 235 empresas que reúnen las características delimitadas en la población de estudio. Procedemos ahora a calcular el tamaño de la muestra:

N ? *Tamaño de la Población* ? 235

p ? *Probabilidad muestral* ? x / n ? 5% ? q ? $1 - p$

e ? *Error Estandar permitido* ? 5%

Z ? *Nivel de Confianza* ? 95% ? z ? 1.96

$$n ? \frac{N * p * q}{z^2 * e^2} ? 55$$

³⁵ Fuente Guía Roji. (2000), Censo de empresas del area metropolitana de la ciudad de Monterrey, México.

Por lo anterior, tenemos entonces que el tamaño óptimo de la muestra con la cual se desarrolló el análisis fue de 55 empresas a un nivel de confianza del 95% y con un error estándar permitido del 5%.

3.8.a Estadística descriptiva de la investigación.

A continuación en las Tablas 3.5 al 3.8, se detallan algunas estadísticas descriptivas del estudio.

Tabla 3.5 al 3.8 Estadística descriptiva del muestreo de las empresas del AMM.

# DE EMPRESAS	CVE.SOL	ND	EXP	PROP	ERP	APS	MOD.OPT	Total general	%
PLAN.RED.LOG			3	2	2			7	13%
PLAN.LAYOUTS					1			1	2%
BAL.CARGAS			1	1		1		3	5%
PROG.DIST			5	1	4			10	18%
PROG.PROD			3	3	14	6	2	28	51%
NO DISPONIBLE		3	1	1	1			6	11%
Total general		3	13	8	22	7	2	55	
%		5%	24%	15%	40%	13%	4%		

# DE EMPRESAS	ESTATUS	NA.	PLAN	PROCESO	ABORT	OK	Total general	%
EXP		7	1				8	21%
PROP			1	1		2	4	11%
ERP			14			4	18	47%
APS			1	4	1		6	16%
MOD.OPT				1		1	2	5%
Total general		7	17	6	1	7	38	
%		18%	45%	16%	3%	18%		

# DE EMPRESAS	ESTATUS	PLAN	PROCESO	ABORT	OK	Total general
ERP		14			4	18
APS		1	4	1		6
Total general		15	4	1	4	24
%		63%	17%	4%	17%	

# de Empresas	TIPO	JOB SHOP	ROUTING	LOTSIZING	FLOW SHOP	Total general
EXPERIENCIA		2	5	1		8
SIST.PROPIETARIO		3	1			4
ERP		7	4	4	3	18
APS		1		3	2	6
MOD.OPT					2	2
Total general		13	10	8	7	38
%		34%	26%	21%	18%	

Tamaño de la Población:
235

Probabilidad de Suceso:
5%
2 Sucesos de 55 = 3.4%

% ERROR TOLERADO:
5%

CONFIABILIDAD:
95% 1.96

Tamaño de la Muestra (n):
55

- ? Tabla 3.5: Estratificación de las empresas de acuerdo al tipo de problema logístico que están enfrentando y de acuerdo al tipo de solución que están llevando a cabo.
- ? Tabla 3.6: Estratificación de las empresas que enfrentan el problema de secuenciación logístico de acuerdo al tipo de solución que están llevando a cabo y su estatus a la fecha del estudio.
- ? Tabla 3.7: Estratificación de las empresas que enfrentan el problema de secuenciación logístico y que

han implementado como solución algún tipo de sistema ERP o APS y de acuerdo al tipo al estatus del proyecto a la fecha del estudio.

? Tabla 3.8: Estratificación de las empresas que enfrentan el problema de secuenciación logístico de acuerdo al tipo de secuenciación que se requiere resolver y a la solución que estuvieron llevando a cabo a la fecha del estudio.

En la tabla anterior se puede apreciar que la cantidad de empresas que afirmaron utilizar como estrategia de solución para sus problemas de logística el desarrollo de modelos matemáticos, sólo representan ser dos empresas. Estas 2 empresas al compararse contra las 55 de la muestra resulta apenas en un 3.4% lo cual comprueba y ratifica el valor de “p” que ya antes de manera previa se había estimado a un 5% al iniciar el experimento ³⁶.

3.8.b Exposición de resultados y conclusiones.

La validez del experimento fue comprobada mediante una prueba de confiabilidad “Alfa Croanbach” mediante la utilización del Software SPSS versión 13.0 con un índice de 0.94 el cual nos calculó los siguientes resultados:

1. El análisis de comunalidad reporta que todas las variables aparecen por arriba de 0.83 por lo que no hay evidencia de que alguna de las variables tenga que ser omitida del instrumento.
2. El Análisis de Varianza Explicada demuestra que las variables del instrumento explican más del 91% de la varianza del instrumento.
3. La matriz de componentes principales (Constructos) evidencia que existe una fuerte correlación entre el “Tipo de operación” de la compañía respecto al “Tipo de problema” que actualmente enfrenta. Así también existe una fuerte correlación entre el “Tipo de solución” implementada en la compañía y el “Estatus actual” que guarda la solución del problema.
4. La Matriz de Factores de Rotación resultó con un índice de 0.197, lo cual es evidencia de que el factor de normalización que tuvo que ser aplicado en los datos fue pequeño.

La conclusión de los análisis aquí presentados, evidencia que el instrumento de medición es confiable y que por tanto la medición de los datos que de aquí se obtenga también lo es. Por tanto, a continuación se describen los resultados de la investigación:

³⁶ Supra, pag 85.

1. La incidencia de las empresas que mencionó a la logística estratégica como parte de su problemática actual es sólo el 13%, es decir únicamente 7 empresas de las 55. En lo que se refiere a la logística táctica ésta se lleva el 7% de incidencia.
2. Por lo anterior se comprueba que en el AMM, el 70% de las 55 empresas revisadas, es decir 38 empresas, declaran enfrentar actualmente problemas relacionados a la logística operativa.
3. De este último 70%, la clasificación a su vez se desdobra de la siguiente manera:
 - a. Programación de Producción discreta "Job Shop Scheduling" (34%)
 - b. Programación de Producción continua "Lotsizing Scheduling": (21%)
 - c. Programación de Producción flexible "Flow Shop Scheduling": (18%)
 - d. Programación de Ruteo "Routing Scheduling" : (26%)
 - e. Programación de Recursos "Workforce Scheduling" : (0%)
4. A partir de este mismo 70% de empresas, se observa también la siguiente clasificación:
 - a. El 5% de las empresas declaró utilizar modelos de optimización como método de solución.
 - b. El 32% de empresas manejan métodos basados en la experiencia o en sistemas propietarios.
 - c. El 63% de las empresas basan su estrategia en el uso de tecnología comprada. A su vez el 80% de estas empresas, están en planes de implementación de un Sistema ERP o APS.
5. De lo anterior resulta que más del 80% de la logística operativa redunda en problemas de Scheduling catalogados como "Job Shop", "Lotsizing" y "Routing".
6. De las 55 empresas de la investigación tenemos que solo el 11% afirmaron NO enfrentar actualmente ningún problema logístico. Esto no es evidencia suficiente de que efectivamente no enfrente actualmente el problema, sin embargo esto rebasa el alcance de la investigación.

Los resultados anteriores obtenidos a partir de una muestra de 55 empresas grandes y medianas de la Ciudad de Monterrey, hacen concluir que son relativamente muy pocas las empresas (7) las que estuvieron dispuestas a invertir en sistemas APS. Más aún, las implementaciones de estos sistemas APS no han logrado aún resolver en términos reales sus problemas de logística.

3.9 Comentarios finales.

A lo largo de este capítulo se describieron y se modelaron en términos matemáticos algunos de los problemas más frecuentemente enfrentados en las áreas de logística. Cada uno de estos problemas contiene sus propias características y propiedades que los hacen interesantes y dignos de exploración en términos de una investigación más exhaustiva. La presente tesis doctoral se concentrará en dar solución a uno de estos problemas, el cual fue nombrado genéricamente como un problema de logística de ruteo de distribución.

La revisión bibliográfica hecha en el apartado 3.6 así como las investigaciones de campo expuestas en el apartado 3.7 evidencian que en realidad lo que más aflora a la superficie en los sistemas del tipo APS es la falta de rigor matemático en sus métodos de optimización. Como remedio a lo anterior, pareciera que estos sistemas APS promovieran la utilización de metodologías basadas en conceptos tales como "Logística Colaborativa". El concepto anterior no es equivocado, sin embargo esto no es optimización sino tan solo se trata de reingeniería.

Desde un punto de vista rigurosamente científico, los algoritmos computacionales incluidos en los sistemas APS, no contienen modelos de optimización basados en motores matemáticos. Cuando mucho presentan indicios de utilizar heurísticas poco sofisticadas para su ejecución. Por tanto, las soluciones que estos ofrecen, no solamente dejan de ser óptimas matemáticamente hablando, sino también pocas veces incluyen las consideraciones que en la práctica se requieren en las empresas.

Así pues al día de hoy, las empresas solo tienen dos alternativas de acción, o siguen "apostando" fuertes sumas de dinero a las soluciones del tipo APS que hay en el mercado tras la promesa de alcanzar la tierra prometida para la solución de la logística, o se deciden a invertir en investigación y desarrollo para generar soluciones propias y especializadas que realmente sean óptimas y resuelvan el problema.

Como se pudo apreciar particularmente en el estudio de campo cubierto en el apartado 3.8, los problemas de optimización de ruteo en las empresas del área metropolitana de la ciudad de Monterrey, caracterizan mas del 25% del tipo de problemas que actualmente se enfrentan en las áreas de logística de las empresas que fueron muestreadas. Por tal motivo consideramos que dicho tipo de problema es digno de ser analizado para ser tratado y resuelto en nuestro proyecto de investigación. Dicha tarea se inicia en el siguiente capítulo de la tesis en la cual se abordarán los fundamentos matemáticos requeridos para atender los problemas de logística de ruteo.

La ciencia ha explorado el microcosmos y el macrocosmos; ahora tenemos una buena idea de la superficie de la tierra.

La gran frontera aún inexplorada es la complejidad.

Heinz Pagels

4. Fundamentos matemáticos para problemas de logística de ruteo.

4.1 Introducción: significado del concepto “problema difícil de resolver” (NP-Hard).

El OR/MS, nació hace ya más de cincuenta años cuando George Dantzig inventó el método Simplex para resolver problemas de optimización lineal, es decir, problemas cuyas variables de decisión son continuas y relacionadas de manera lineal. Aun cuando en sus orígenes, esta nascente área de la ciencia fue motivada por aplicaciones de carácter militar, el OR/MS fue alcanzando un alto grado de interés entre investigadores y profesionistas en los campos de ingeniería, matemáticas aplicadas y administración, quienes motivados por los diversos y complejos problemas de toma de decisiones que surgían en varias áreas del quehacer científico e industrial, comenzaron a estudiar y desarrollar metodologías de solución para problemas de diferentes características. Fue así como nacieron posteriormente otras ramas de la optimización tales como:

- a. Optimización NO lineal: manejo de relaciones no lineales entre las variables de decisión.
- b. Optimización discreta: variables enteras y/o binarias para el manejo de condiciones en las cuales se requiere restringir y/o explotar ciertas propiedades de los problemas en cuestión.
- c. Optimización entera mixta: manejo de variables continuas y discretas.

El OR/MS se encuentra en prácticamente todos los tipos de industrias. Es evidente que las corporaciones aspiran a tomar decisiones que les reditúen en beneficios económicos, y normalmente, estas decisiones se encuentran restringidas de forma muy compleja. Estos atributos son únicos de los modelos de OR/MS. “En las últimas décadas el impacto del OR/MS en la industria ha sido impresionante, convirtiéndose en ganancias con frecuencia multimillonarias en los diversos ramos industriales”³⁷.

Establecer cuándo un problema es “fácil” o “difícil”, equivale a determinar cuándo un problema es NP-Hard y cuándo no. Determinar lo anterior está íntimamente ligado al tiempo computacional requerido

³⁷ Ríos, Roger. (1999), Aplicaciones del TSP, “Ingenierías 2(4)”, México, pag 18-23.

para la solución de cada problema en cuestión. Sin entrar en detalles técnicos, podemos afirmar que un problema es “fácil” de resolver, es decir no es un problema NP-Hard, cuando es posible encontrar un algoritmo o método de solución, cuyo tiempo de ejecución computacional crece de forma moderada (o polinomial) respecto al tamaño del problema. Por el contrario, si no existe tal método de solución o algoritmo que pueda resolver el problema en un tiempo polinomial, entonces decimos que el problema es “difícil” de resolver o NP-Hard.

El término utilizado en Complejidad Computacional denominado “NP-Hard” ha sido siempre visto como el icono y la cima de la inherente intratabilidad de los modelos matemáticos que los investigadores crecientemente afrontan durante el desarrollo de sus algoritmos para resolver problemas complejos. Probar la complejidad de un problema, es decir probar si un problema es NP-Hard, puede llegar a ser tan complejo como al igual resulta tratar de encontrar algún algoritmo eficiente para resolver el problema en sí. Descubrir que un problema en particular es “NP-Hard” es usualmente el inicio de cualquier trabajo de investigación dedicado a resolverlo. Dicho de otro modo, esta primera fase de la investigación proporciona información valiosa acerca de cual enfoque o estrategia sería la más adecuada para resolver el problema en cuestión.

El hecho de que un problema sea NP-Hard, no implica que el problema no pueda resolverse, sino que cada algoritmo existente para la solución del problema tiene un tiempo de ejecución que crece explosivamente (o exponencialmente) de acuerdo al tamaño del problema que se esté requiriendo resolver. La consecuencia directa de un algoritmo que para su ejecución computacional, tiene una función de tiempo exponencial es que a medida que aumenta el tamaño del problema, el tiempo requerido para la solución aumenta de forma exponencial, lo cual limita bastante el tamaño de los problemas que pueden resolverse en la práctica.

Matemáticamente hablando, determinar si un problema es NP-Hard o no se denomina establecer la complejidad computacional del problema. El campo de la “Complejidad Computacional” es toda una disciplina por separado cuyo propósito especial es específicamente demostrar cuando un problema es NP-Hard. El “Problema del Agente Viajero” (o TSP), es quizás el problema clásico del tipo NP-Hard para el cual mayor investigación se ha desarrollado a lo largo de la historia. El TSP como tal, tiene muy diversas aplicaciones para resolver problemas de secuenciación de tareas en manufactura así como problemas ruteo de vehículos en el ramo de la logística. El TSP tiene la particularidad de que es un problema relativamente sencillo de formular pero por otro lado es muy difícil de resolver (NP-Hard).

Aunque será hasta el siguiente punto cuando se abordará de lleno la definición del TSP, con la finalidad de comprender aún mejor la naturaleza de los problemas tipo NP-Hard; supongamos entonces una instancia del TSP con “n” nodos. Una forma poco eficiente de resolver el problema, sería por enumeración exhaustiva. Es decir, probar todas las posibles combinaciones de soluciones que pueda tener el problema en cuestión y al revisar cada una de las soluciones, pues simplemente se escoge la mejor de todas ellas.

Para nuestro caso tendríamos que procesar $(n-1)!$ combinaciones para recorrer todas las posibilidades de manera enumerativa. Esto resulta en un esfuerzo computacional enorme ya que $n! = n(n-1)(n-2)\dots(2)(1)$. Para cada una de estas $(n-1)!$ combinaciones habría entonces que calcular la solución del problema, eligiendo de entre todas estas combinaciones aquella solución que sea la mejor.

Para el procedimiento antes explicado (enumerativo), decimos que el problema ha quedado totalmente resuelto ya que hemos procesado todas las combinaciones posibles. El tiempo de ejecución de este algoritmo decimos que crece de acuerdo a una función del tipo $f(n)=(n)!$ (función de crecimiento exponencial). Esta función de crecimiento puede comprenderse que es “no polinomial” y por tanto deja de ser viable su implementación computacional sobre todo en el caso de instancias grandes. Por ejemplo, en el caso de que tengamos una red con 5 nodos ($n=5$), tendríamos que calcular $4!$, lo que equivale a 24 combinaciones. Lo anterior en realidad no es un impedimento en términos computacionales. Sin embargo al considerar una red de 50 nodos ($n=50$), el número posible de tours es igual a $49!$, el cual es un número tan gigantesco que no alcanzaría a resolverse en varios meses ni en la computadora más potente del mundo. Cabe mencionar que una red de 50 nodos, apenas si se considera un problema de tamaño mediano.

Aunque la función factorial $f(n)=n!$, se define como una función que crece exponencialmente a medida que crece el valor de “n”, lo anterior, no resulta evidencia suficiente para comprobar que el TSP sea un problema NP-Hard, ya que quizás pudiera existir algún algoritmo que lo pudiera resolver en un tiempo de ejecución polinomial. En este caso, sin embargo, ya se ha demostrado que tal algoritmo con comportamiento polinomial no existe y que por lo tanto irreductiblemente el TSP pertenece a la clase de problemas NP-Hard³⁸. Con el fin de dimensionar la diferencia entre la naturaleza de una función

³⁸ Garey, D. (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, New York, pag 230-242.

con crecimiento polinomial y una con crecimiento exponencial, a continuación se muestra la tabla 4.1 extraída de Garey. Las cifras que se muestran, representan el tiempo de procesamiento que ocuparía una computadora que pudiera procesar 1 millón de operaciones de punto flotante por segundo. Nótese el crecimiento explosivo (exponencial) que se tiene en las últimas dos columnas de la tabla 4.1, lo anterior muestra el incremento sustancial de tiempo requerido (años y siglos) para la obtención computacional de la solución óptima.

Tabla 4.1 Tiempo computacional de acuerdo al grado de complejidad de un problema ³⁹.

Tamaño <i>n</i>	$f(n)=n$	$f(n)=n^2$	$f(n)=n^3$	$f(n)=n^5$	$f(n)=2^n$	$f(n)=3^n$
10	.00001 seg	.0001 seg	.001 seg	.1 seg	.001 seg	.059 seg
20	.00002 seg	.0004 seg	.008 seg	3.2 seg	1.0 seg	58 minutos
30	.00003 seg	.0009 seg	.027 seg	24.3 seg	17.9 minutos	6.5 años
40	.00004 seg	.0016 seg	.064 seg	1.7 minutos	12.7 días	3855 siglos
50	.00005 seg	.0025 seg	.125 seg	5.2 minutos	35.7 años	2×10^8 siglos
60	.00006 seg	.0036 seg	.216 seg	13 minutos	366 siglos	1.3×10^{13} siglos

Como complemento a lo expuesto anteriormente en la Tabla 4.1, a continuación en la siguiente hoja, se muestra en la figura 4.1 una breve explicación acerca de que es lo que significa el que un problema sea nombrado como “NP-Hard”. Aunque una de las características de este tipo de problemas es que su espacio solución es enorme, no obstante la figura 4.1 lo que busca ilustrar es lo “intrincado” que resulta éste espacio, al igual que la falta de información necesaria para poder explorarlo eficientemente. Así entonces hablar de un problema “NP-Hard”, equivale a decir que éste es intratable a través de los medios matemáticos convencionales.

³⁹ Ríos, Roger. (1999), Aplicaciones del TSP, “Ingenierías UANL 2(4)”, México, pag 18-23.

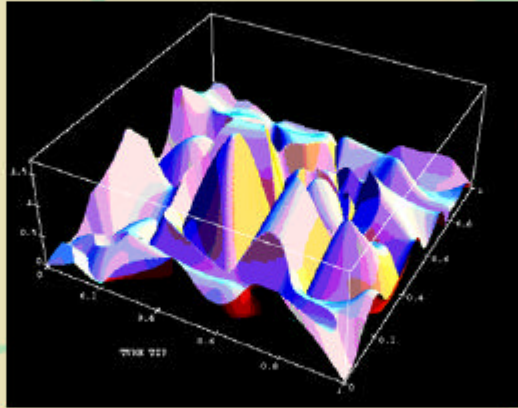
Porqué se aplican los AG en los NP-Hard ?

Con una computadora que pueda procesar 1000 millones de Comb/segundo tomaría:
8 mil millones de millones de años

Por ejemplo, una red con 30 nodos tendría 30! combinaciones:
 2.65×10^{32}

Consiste en encontrar la ruta más corta que haga visitar "n" cantidad de nodos

El espacio de búsqueda es sumamente grande y complejo imposibilitando el uso de las matemáticas convencionales



Un problema famoso tipo NP-Hard es el problema del Agente Viajero (TSP)

Se dispone de poco conocimiento del espacio solución que permita delimitarlo.

La "campaña" es topográficamente compleja y llena de puentes, túneles y pasajes "intrincados"

Encontrar el camino correcto redunda en la intratabilidad de los problemas NP-Hard a través de las matemáticas.

Figura 4.1 Intratabilidad matemática del TSP.

4.2 Introducción al problema del agente viajero (TSP).

Ya antes en la figura 4.1, se expuso que uno de los problemas más famosos y difíciles en la teoría de optimización es el problema del agente viajero o también denominado (Traveling Salesman Problem o TSP). Para el problema del TSP, han sido la simplicidad de su formulación aunado a la dificultad para resolverlo, los factores que atrajeron a investigadores en las áreas de matemáticas discretas en todo el mundo a estudiar el problema. Aunado a lo anterior, el interés en el estudio de técnicas para su solución es motivado por la enorme cantidad de aplicaciones prácticas de problemas de toma de decisiones donde éste aparece como subestructura. Se hará una breve introducción al igual que algunas de sus aplicaciones prácticas.

El problema del TSP se formula de la siguiente manera: un agente viajero, partiendo de su ciudad de origen, debe visitar exactamente una vez cada ciudad de un conjunto de ellas previamente especificadas y retornar al punto de partida⁴⁰. Un recorrido con estas características, es llamado dentro de este contexto un “tour”. El problema consiste en encontrar el tour para el cual la distancia total recorrida sea mínima. Se asume que se conoce, para cada par de ciudades, la distancia entre ellas. La Figura 10 ilustra un tour en una instancia de ocho ciudades. Representada por un grafo, donde cada nodo del grafo corresponde a una ciudad y cada arco que une a un par de nodos representa la parte del tour que pasa por dichos nodos. En la Figura 4.2, se ilustra el tour que visita las ciudades 1, 2, 3, 8, 5, 4, 7, 6 y 1, en ese orden.

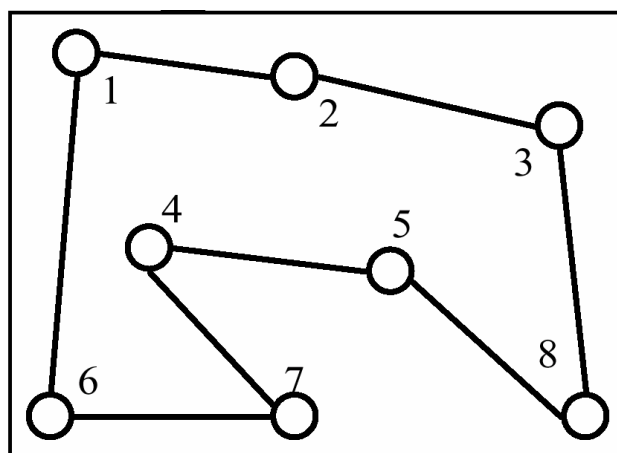


Figura 4.2 Un tour en un TSP de ocho ciudades

⁴⁰ Lenstra, J. (1985), The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization, Editorial Wiley, Chichester England, pag 30-45.

En la solución computacional de los problemas como el TSP, es importante evaluar las ventajas y desventajas que surgen entre la obtención de soluciones de alta calidad contra los recursos y tiempo computacional empleados para obtenerla. Como ya antes se comentó, el problema del TSP en sí es fácil de formular, sin embargo, al igual que muchos otros que se presentan en el campo de optimización discreta (o también optimización combinatoria), es sumamente difícil de resolver (NP-Hard). Cuando decimos “resolver”, nos referimos a encontrar la solución óptima al problema y a probar que ésta es efectivamente la mejor solución posible.

Para 1990 y tomando en consideración los antecedentes revisados, la instancia del TSP de mayor tamaño que había sido resuelto era una de 318 nodos. Hoy en día existen métodos basados en técnicas de ramificación y corte/acotamiento, las cuales han obtenido provecho al explotar efectivamente la estructura matemática del problema. En 1998, se reportó la instancia del TSP más grande que se ha resuelto hasta nuestros días, estamos hablando de una red de 13,509 nodos la cual fue resuelta mediante la aplicación de computadoras interconectadas con procesamiento paralelo⁴¹. El tiempo requerido para dicha solución no es mencionado en la obra antes citada, lo que si es previsible es que dicho tiempo haya sido significativamente menor en términos relativos por haberse tratado de un experimento con procesadores computacionales en paralelo.

A pesar de que lo anterior evidencia el tremendo progreso logrado durante la década de los noventa, es importante precisar en este momento que el caso de nuestro proyecto de investigación no busca enfrentar el reto algorítmico y computacional para solucionar problemas del TSP de gran envergadura (por ejemplo arriba de 1000 nodos), sino más bien se concentra en dar tratamiento a una variante más sofisticada perteneciente a la familia de los problemas relacionados al TSP. Dejaremos la exposición de dicha variante para el capítulo 6.

⁴¹ Optima. (1998), Mathematical Programming Society Newsletter, EUA, pag 58.

4.3 Aplicación del TSP en problemas generales de secuenciamiento en manufactura.

Existen razones prácticas que hacen importante al problema del TSP. Muchos problemas reales en el área de manufactura y de logística pueden formularse como instancias del TSP. Por ejemplo el problema de secuenciamiento para la programación de tareas en una máquina. Supongamos entonces un taller de manufactura en el cual se cuenta con una sola máquina en la cual es posible procesar diferentes tareas, una a la vez.

Ahora bien, para procesar cada una de estas tareas, la máquina requiere de cierta configuración característica de la tarea, pueden ser: número y tamaño de diferentes dados, colocación de cuchillas a cierta distancia unas de otras, colorantes para alguna fibra, etc. De manera que una vez que una tarea ha sido terminada, es necesario preparar la máquina para procesar una nueva tarea, aquí será necesario invertir un cierto tiempo, y este tiempo dependerá de la tarea recién procesada así como de la siguiente tarea a procesar. Si las características de una tarea son similares a las de otra, es adecuado pensar que el tiempo que se requiere para pasar de una configuración a otra será pequeño, en comparación con el tiempo que sería requerido para pasar de una tarea a otra con características o configuraciones de fabricación muy diferentes.

La problemática aparece al percatarnos que durante las labores de preparación de la máquina, ninguna de las tareas se puede ejecutar. Así, este tiempo correspondiente al recurso productivo en cuestión se pierde inevitablemente provocando como consecuencia un desaprovechamiento en la capacidad de manufactura, no solo de la máquina, sino también de todo el resto de los recursos que se dejan de aprovechar durante los tiempos de preparación antes mencionados. Lo anterior, representa un costo de oportunidad para la empresa. Por tanto, el problema consiste entonces en encontrar el orden o la secuencia en que se deben procesar las tareas con la finalidad de minimizar el tiempo total que transcurre en el sistema en el cual se están ejecutando labores de preparación.

Así pues, aunque este problema de secuenciamiento aplicado para un ambiente de manufactura parezca no tener ninguna relación con el problema del TSP, existen ciertas similitudes que permiten que se pueda formular matemáticamente de la misma manera. Es decir, cada una de las tareas a ser procesadas en la máquina puede ser vista como cada uno de los nodos que se requieren visitar en la red. Por otro lado, el tiempo necesario para cambiar la configuración de la máquina entre una tarea y otra corresponde a la distancia que hay que recorrer para trasladarse de un nodo al siguiente durante la

conformación del tour. Entonces encontrar la manera de ordenar las tareas para minimizar el tiempo total de preparación equivale en un problema de ruteo a encontrar la secuencia de los nodos a ser visitados en una red para formar la ruta óptima que minimice la distancia total recorrida. Lo anterior nos debe ofrecer una buena perspectiva de lo crucial que resulta tener buenos algoritmos para la solución de problemas para el TSP, inclusive aplicado para ambientes de manufactura.

4.4 Clasificación de los métodos propuestos para solución del TSP.

Como ya se expuso en el apartado anterior, la implicación directa de un problema NP-Hard es que cualquier algoritmo empleado para encontrar su solución óptima emplea un tiempo de cómputo que crece exponencialmente con el tamaño de la instancia del problema. Por tal motivo, se origina la necesidad de emplear heurísticas, las cuales son procedimientos que aunque no garantizan una solución óptima al problema, si ofrecen una solución factible relativamente cercana al óptimo y en un tiempo de ejecución computacional razonable⁴².

Los problemas “NP-Hard” en las últimas tres décadas han sido muy estudiados. Se han explorado múltiples técnicas para la búsqueda de algoritmos eficientes en la solución de este tipo de problemas. Básicamente las técnicas de solución pudieran ser clasificadas en 3 grandes rubros: métodos enumerativos, métodos exactos y métodos heurísticos.

Los métodos enumerativos o también denominados de “fuerza bruta”, como su nombre lo indica enumeran el conjunto de combinaciones numéricas explícitamente, analizando el total o al menos una parte del espacio de búsqueda. En cada combinación o enumeración se requiere verificar el cumplimiento de las restricciones para asegurar cuando existe realmente una solución y a su vez cuando ésta supuesta solución alcanza el punto de optimalidad respecto a las soluciones previamente alcanzadas. La desventaja de estos métodos es que pueden resultar extremadamente lentos para la exploración del espacio solución. Aún con la tecnología computacional actual se pudieran requerir siglos para la solución de problemas apenas de tamaño típico.

El segundo grupo corresponde a los métodos exactos. Estos métodos son mucho más inteligentes que los de “fuerza bruta” descritos en el primer grupo. Los métodos de este tipo son aquellos normalmente utilizados en Programación Lineal (LP), Programación NO Lineal (NLP), Programación Entera (IP), Programación Binaria (BP) y por supuesto en la Programación Mixta (MIP). Aún este tipo de métodos, cuando son aplicados para el mismo grupo de problemas mencionados antes, pueden quizás no requerir siglos, pero si acaso meses o años en términos de esfuerzo computacional para resolverlos. Como es de imaginar, en la práctica logística convencional no es posible esperar a veces ni siquiera horas antes de

⁴² Goldberg, David. (1989), Genetic Algorithms in Optimization and Machine Learning, Addison-Wesley Publishing Company, EUA, pag 412.

requerir estar tomando una decisión para un problema típico. Los métodos de solución exacta, por definición tienen el objetivo de encontrar la solución óptima verdadera para los problemas de optimización. Algunos de estos métodos serán revisados más adelante en el apartado 4.8.

El tercer grupo corresponde a las técnicas heurísticas. El concepto “Heurística”, proviene del griego “euriskein” que significa encontrar ó descubrir. El Matemático Arquímedes la utilizó por 1era vez: cuando el utilizó la palabra “Eureka”. Los métodos heurísticos, son algoritmos que sin ofrecer una garantía de éxito, aproximan buenas soluciones en tiempos razonables para problemas que normalmente no pueden ser tratados mediante métodos enumerativos⁴³.

Estos métodos son muy útiles en la actualidad, pues a pesar de que sólo se obtienen soluciones satisfactorias y casi nunca la solución óptima exacta, los mismos son muy efectivos para problemas de gran tamaño y complejidad. La clave consiste en elegir convenientemente el tipo de heurística, de acuerdo al tipo de problema que se plantee resolver. A este último respecto, podemos añadir que básicamente existen dos grandes clases de heurísticas. La primera clase corresponde a aquellas heurísticas que son especializadas de acuerdo al tipo de problema. Así por ejemplo en el siguiente apartado 4.5 se expondrán algunas de estas en lo que se refiere a aplicaciones especializadas para el problema del TSP.

La segunda clase corresponde a las heurísticas de propósito general o también llamadas meta-heurísticas. Entre las técnicas meta-heurísticas más utilizadas en la actualidad se mencionan las siguientes sólo con la finalidad de situar al lector en el tema:

1. Búsqueda tabú: basado en el psicoanálisis freudiano, el cual se ampliará en el apartado 4.6
2. El algoritmo genético: proveniente de la evolución biológica.
3. El enfriamiento simulado: proveniente del campo de la mecánica estadística.
4. La colonización de hormigas: proveniente del mecanismo mediante el cual las hormigas buscan su alimento.

El método basado en los “algoritmos genéticos” será la meta-heurística propuesta para el problema de investigación, la cual será desarrollada ampliamente en el capítulo 5. A continuación en los apartados 4.5 al 4.8 se exponen algunos de los métodos mencionados hasta aquí. En el apartado 4.9 se resumirán los métodos mencionados a través del uso de una tabla esquemática.

⁴³ Greenberg , H. (1996), *Mathematical Programming Glossary*, disponible en <http://www.cudenver.edu/~hgreenbe/glossary>.

4.5 Revisión de métodos basados en heurísticas para la solución del TSP.

Las heurísticas de propósito especial que han sido propuestas para resolver el TSP se han denominado así debido a que explotan la estructura y las características particulares del problema. Una heurística que ha sido particularmente estudiada para este tipo de problemas es la heurística de tipo miope (o Greedy en inglés). Es llamada así porque en su estrategia de búsqueda sólo se intenta mejorar la solución actual sobre la base de una exploración de otras alternativas dentro del vecindario local, sin buscar más allá de un cierto entorno muy cercano. Dentro de estas heurísticas existen algunas variantes las cuales irremediamente se desarrollan a partir de la explotación de las características particulares de cada tipo de problema en cuestión.

1. Heurística del vecino más cercano: esta heurística se emplea en la fase de construcción del tour o solución inicial. Por tratarse de un procedimiento constructivo, se parte al elegir un nodo inicial, llamémoslo j_1 . Una vez seleccionado, se mide la distancia que hay de este nodo a los restantes, y a continuación la heurística indica ahora elegir aquél otro nodo cuya distancia al nodo inicial sea la mínima, es decir, elegimos al vecino más cercano. Este nodo agregado se le llama j_2 . A partir de aquí la heurística se va aplicando de manera constructiva hasta lograr alcanzar una trayectoria de abarque todos los nodos de la red ($j_1, j_2, j_3, \dots, j_k, j_{k+1}, \dots, j_n$), donde el nodo j_{k+1} se elige tomando la mínima distancia que hay desde j_k hasta cada uno de los nodos que sean distintos de los ya elegidos durante el procedimiento recursivo.

Al terminar con el último nodo, se agrega un último arco que va del nodo “ j_n ” hasta el nodo “ j_1 ” para entonces finalizar cerrando el tour por completo. Esta heurística es muy eficiente computacionalmente hablando en las primeras iteraciones, sin embargo, la desventaja que presenta es que en las últimas fases de la construcción del tour, el procedimiento puede verse obligado a tener que elegir arcos cuya longitud sea particularmente grande, especialmente en la última que va desde j_n a j_1 .

2. Heurística de inserción más cercana: este procedimiento también es constructivo, pero en contraste con el anterior, en el cual se tiene sólo un camino, y sólo hasta el final se completa un tour; aquí tenemos subtours, los cuales van creciendo hasta completar un tour que abarque todos los nodos. El algoritmo inicia con un subtour denominado T . A partir de aquí se busca insertar el nodo “más cercano” a este subtour con la intención de ampliarlo. Para lograr lo anterior el procedimiento requiere examinar primero todos los nodos j que no estén aún incluidos en T . De lo anterior se calcula entonces

la distancia de cada uno de estos nodos hacia T lo cual llamaremos $d(j, T)$. Así pues, $d(j, T)$ es la distancia mínima que hay desde el nodo j a cualquiera de los nodos que pertenecen a T. A continuación se ordenan las $d(j, T)$ en forma ascendente. Así, llamemos " j^* " al nodo que se encuentra al principio de esta lista, es decir, al nodo "más cercano" a T.

A partir de aquí se selecciona dentro del subtour T al nodo que se encuentre "más cerca" de j^* , esto es, medimos la distancia desde j^* a cada uno de los nodos que actualmente conformen al subtour T. Llamaremos k^* a aquel nodo dentro de T, cuya distancia a j^* sea la menor de todas. Finalmente el procedimiento de inserción logra ampliar el subtour insertando a j^* entre k^* y alguno de sus dos nodos vecinos que estén contiguos dentro del subtour T. Para lograr establecer cuál de los dos nodos contiguos (k_1 y k_2) es el que resulta más viable para aplicar el procedimiento de inserción, entonces se calcula la distancia que hay de (k_1 a k^*) y de (k^* a k_2). Ambas distancias son de hecho dos arcos que actualmente se encuentran incluidos dentro del subtour T. Si el arco de j^* a k_1 , es menor o igual que el arco de j^* a k_2 , entonces j^* se inserta entre k_1 y k^* . En caso contrario se inserta j^* entre k_2 y k^* . Así pues, el proceso termina una vez que se haya construido un tour completo.

En ambas heurísticas revisadas antes, no se puede garantizar que se produzca una buena solución. Es decir, no existe una garantía de desempeño en la calidad de la solución. Para explicar a que nos referimos con una garantía de desempeño, utilizaremos un ejemplo particular de un problema al cual denotaremos por "I". A partir de aquí $A(I)$ será entonces el valor producido por la heurística o el algoritmo que estemos usando para resolver el problema "I". Si ahora entonces definimos que $OPT(I)$ sea el valor de la solución óptima para "I" entonces tenemos que $OPT(I)$ debe de ser menor o igual que la longitud de cualquier otro tour, por lo que $OPT(I) \leq A(I)$. Luego entonces el algoritmo de aproximación tiene una garantía de comportamiento c^* , donde c^* es un número real, si para cualquier instancia del problema I, se puede probar que $A(I) \leq c^* OPT(I)$.

Este valor c^* indica que si su valor es 1, entonces el algoritmo de aproximación siempre producirá la solución óptima, ya que combinando las dos desigualdades, se tiene que $OPT(I) = A(I)$. Por otra parte, su valor no puede ser menor que 1, ya que en ese caso, se tendría que $A(I) < OPT(I)$, es decir el algoritmo de aproximación produciría un valor mejor que el óptimo, lo cual es imposible. Así que c^* tiene que ser un valor mayor o igual que 1. Ahora bien, mientras más cerca se encuentre de c^* respecto a la unidad (1), entonces tenemos que el algoritmo de aproximación, obtendrá soluciones que se encuentran cerca del valor óptimo. Por el contrario, si este valor es grande, esto indica que se pueden

producir soluciones alejadas del valor óptimo.

Aunque en el caso del TSP no se tiene un algoritmo que ofrezca una garantía para la obtención de una solución exacta (óptima), es posible tener algoritmos que ofrezcan garantías de desempeño en términos predecibles. Para esto la condición particular que se requiere por parte de estos algoritmos es que las instancias examinadas posean la propiedad de desigualdad del triángulo. Esta propiedad puede describirse de la siguiente manera: para viajar de un nodo a otro es más corto hacerlo directamente que pasando por un nodo intermedio. Más formalmente, se tiene que cumplir que la distancia del nodo “i” al nodo “j” para cualquier par de nodos debe de ser menor o igual a la distancia requerida para ir del nodo “i” a nodo “k” más la distancia para ir del nodo “k” al nodo “j”.

Si la propiedad anterior se cumple para todos los nodos de la red, entonces sí es posible dar una garantía de desempeño. Por ejemplo aplicando lo antes mencionado para la heurística del vecino más cercano, podemos definir $NN(I)$ como la solución ofrecida por dicha heurística. Entonces es posible demostrar que para cualquier instancia del problema I con n nodos, se tiene un valor de c^* que puede garantizar un desempeño respecto a la solución óptima. El valor que guarda c^* respecto a $NN(I)$ y a $OPT(I)$ se define a través de la siguiente desigualdad ⁴⁴:

$$NN(I) \leq \frac{1}{2} (\log 2 n + 1) * OPT(I)$$

Por otro lado, para valores arbitrariamente grandes de n (instancias del TSP de tamaño grande), la función que expresa la garantía de desempeño de acuerdo a Christofides se define por:

$$NN(I) > \frac{1}{3} (\log 2 (m + 1) + \frac{4}{3}) OPT(I)$$

Ambas desigualdades, lo único que evidencian es que las soluciones ofrecidas por la heurística del vecino más cercano, dejan mucho que desear. En la primera desigualdad no tenemos un valor constante que nos dé una garantía para todas las instancias, ya que c^* está definida por una función exponencial así que en todo caso podríamos decir que $c^* = ?$. Luego la segunda desigualdad nos asegura que encontraremos instancias para las cuales la heurística produce valores muy alejados de la solución óptima.

⁴⁴ Christofides, N. (1976), Worst-case analysis of a new heuristic for the traveling salesman problem, "Report 388 Graduate School of Industrial Administration", Carnegie-Mellon University, Pittsburgh EUA, pag 121-140.

De las heurísticas desarrolladas para el problema del TSP tenemos el trabajo hecho por Christofides como aquel algoritmo que mejor garantía de desempeño ofrece⁴⁵. El trabajo de Christofides combina varias heurísticas, entre ellas las que hemos revisado al principio del presente tema, con la finalidad de construir un tour óptimo. No obstante ser la mejor heurística conocida para el TSP, su garantía de desempeño apenas si logra alcanzar un valor de $c^* = 3/2$. La implicación de este valor para c^* es que, cualquier solución que construyamos con este esquema de aproximación, nos asegurará apenas un valor que nunca excederá en 50 % al valor de la solución óptima.

Algo muy interesante es que si eliminamos esta propiedad de la desigualdad del triángulo, resulta entonces que es imposible obtener un algoritmo de aproximación cuyo comportamiento de tiempo computacional sea polinomial y que tengan una garantía de desempeño. Es decir, si tal construcción fuese posible, equivaldría a entonces decir que existe un algoritmo polinomial que resuelve en forma exacta el TSP. Por el motivo Christofides concluye en su obra que el TSP, desde su variante más básica, es ineludiblemente un problema tipo NP-Hard⁴⁶.

Tomando en cuenta la implicación de que la heurística de Christofides apenas nos asegura una solución que nunca excederá en 50% al valor de la solución óptima, es previsible que sea necesaria aplicar alguna otra estrategia de solución para el problema del TSP. Por tanto, pasamos ahora al apartado 4.6 en el cual revisaremos los fundamentos de la meta-heurística “tabu -search”.

⁴⁵ Ríos, Roger. (1999), Aplicaciones del TSP, “Ingenierías UANL 2(4)”, México, pag 18-23.

⁴⁶ Christofides, N. (1976), Worst -case analysis of a new heuristic for the traveling salesman problem, “Report 388: Graduate School of Industrial Administration”, Carnegie-Mellon University, Pittsburgh EUA, pag 10-12.

4.6 Revisión de la meta-heurística “tabu-search” para la solución del TSP.

A continuación hablaremos de una de las metaheurísticas revisadas provenientes de investigaciones previas relacionadas con la solución del TSP. En principio, las metaheurísticas tiene su origen como métodos de aproximación para atacar particularmente problemas NP-Hard para los cuales las heurísticas de propósito especial, revisadas en el apartado anterior, han fracasado en dar resultados efectivos. Prueba de lo anterior puede ser verificado a partir del valor c^* calculado para la heurística de Christofides expuesto en el apartado 4.5.

Las meta-heurísticas proporcionan marcos generales que permiten generar enfoques híbridos para la solución de los problemas NP-Hard. La aplicación de los Algoritmos Genéticos para el problema del TSP será tratada de manera extensa en el capítulo 5 puesto que la propuesta doctoral se apoya en esta técnica. En este momento revisaremos las propuestas desarrolladas por otros trabajos de investigación en lo referente a la aplicación de la meta-heurística “Búsqueda Tabú”.

Los orígenes de la búsqueda tabú se ubican a fines de los 60s y principios de los 70s, y se atribuyen a Fred Glover. La búsqueda Tabú o también (Tabu Search), está basada en el psicoanálisis freudiano y surgió como un dispositivo que permitiría implementar una estrategia para resolver problemas de optimización combinatoria. La búsqueda tabú puede verse como una meta-heurística que se superpone o combina con cualquier otro mecanismo de búsqueda. La contribución esencial de la “Búsqueda Tabú” es evitar que el proceso de búsqueda al cual está apoyando, quede atrapado en un óptimo local a través de mecanismos de prohibición (o penalización) de ciertos movimientos⁴⁷.

El propósito de clasificar los movimientos como prohibidos (o "tabú") es para evitar que se caiga en ciclos durante el proceso de búsqueda. Los movimientos que se consideran prohibidos constituyen generalmente una pequeña fracción del total de movimientos disponibles, y un movimiento pierde su status de prohibido después de un período de tiempo relativamente corto, volviéndose después nuevamente accesible.

Así pues, desde la perspectiva proveniente del psicoanálisis freudiano, la búsqueda tabú trata de emular el comportamiento de una persona. Es bien sabido que los humanos poseemos un avanzado mecanismo

⁴⁷ Glover, Fred. (1990), Tabu Search: A Tutorial, "Interfaces, Vol 20, No. 4", EUA, pag 74-94.

de intuición que nos permite operar a la vez con información mínima o también cuando la cantidad de datos es muy extensa. Indistintamente solemos introducir un elemento probabilístico en las decisiones que tomamos, lo cual promueve un cierto nivel de "inconsistencia" en nuestro comportamiento. La tendencia resultante en estos casos suele desviarnos de una cierta trayectoria preestablecida, lo cual algunas veces puede ser una fuente de errores, pero en otras ocasiones puede llevarnos a una solución mejor. La búsqueda tabú intenta emular este mecanismo fundamental de la ingenuidad humana.

La Búsqueda Tabú, mantiene información referente a los movimientos más recientes a fin de evitar que una cierta trayectoria previamente recorrida se repita. Para lograr lo anterior se apoya en 3 fundamentos principales ⁴⁸:

1. El uso de estructuras de memoria, diseñadas para permitir una mejor explotación de los criterios de evaluación y búsqueda histórica.
2. Un mecanismo de control para emplear las estructuras de memoria, basado en la interacción entre las condiciones que limitan o flexibilizan el proceso de búsqueda. Este mecanismo se encuentra inmerso en la técnica en forma de reglas que permiten que un movimiento pierda su status de "tabú" debido a que proporciona una mejor solución que la actual.
3. La incorporación de memorias de diferente duración (de corto a largo plazo), para implementar estrategias que intensifiquen (refuercen) las combinaciones que han demostrado históricamente ser buenas; pero también por el otro lado, permitan diversificar la búsqueda hacia nuevas regiones del espacio de soluciones factibles.

Estos dos últimos mecanismos son muy similares a la cruce y la mutación que más adelante se revisarán en el caso de la meta-heurística basada en los Algoritmos Genéticos. El primero nos permite delimitar una cierta región del espacio de búsqueda, mientras que el segundo nos permite saltar a nuevas regiones del mismo, evitando que quedemos atrapados en un óptimo local.

Las estructuras de memoria en la Búsqueda Tabú constituye n una forma de exploración cuyo objetivo es realizar el mejor movimiento posible sujeto a las restricciones que evitan repetir aquellos movimientos que ya han sido revisados (por ser "prohibidos"). El objetivo primordial de los movimientos prohibidos es hacer que la técnica de búsqueda pueda ir más allá de los puntos de

⁴⁸ Glover, Fred. (1993), A user's guide to tabu search, "Annals of Operations Research, Vol. 41", EUA, pag 3-28.

optimalidad local. En general, las restricciones Tabú tienen como objetivo prevenir ciclos e inducir a la búsqueda a que siga una nueva trayectoria.

Algorítmicamente hablando, la Búsqueda Tabú, inicia desde una solución arbitraria. A continuación, el procedimiento se enfoca en la exploración de una vecindad previamente definida para cada punto del espacio solución y elige una nueva solución dentro de tal vecindad, la cual en todo momento busca mejorar el valor que se tiene actualmente como la mejor solución encontrada. El algoritmo concluye una vez que se alcanza una solución tal que es la mejor dentro de la vecindad predefinida a ser explorada. A esta solución alcanzada dentro de la vecindad se le denomina “mínimo local”. Es relevante mencionar que solo en pocas ocasiones, este mínimo local será la solución óptima global del problema. De hecho resulta muy probable que esta solución a la cual se arriba mediante el algoritmo antes descrito quede lejos de la solución óptima global del problema. El funcionamiento básico se ilustra detalladamente en la Figura 4.3 en la siguiente página.

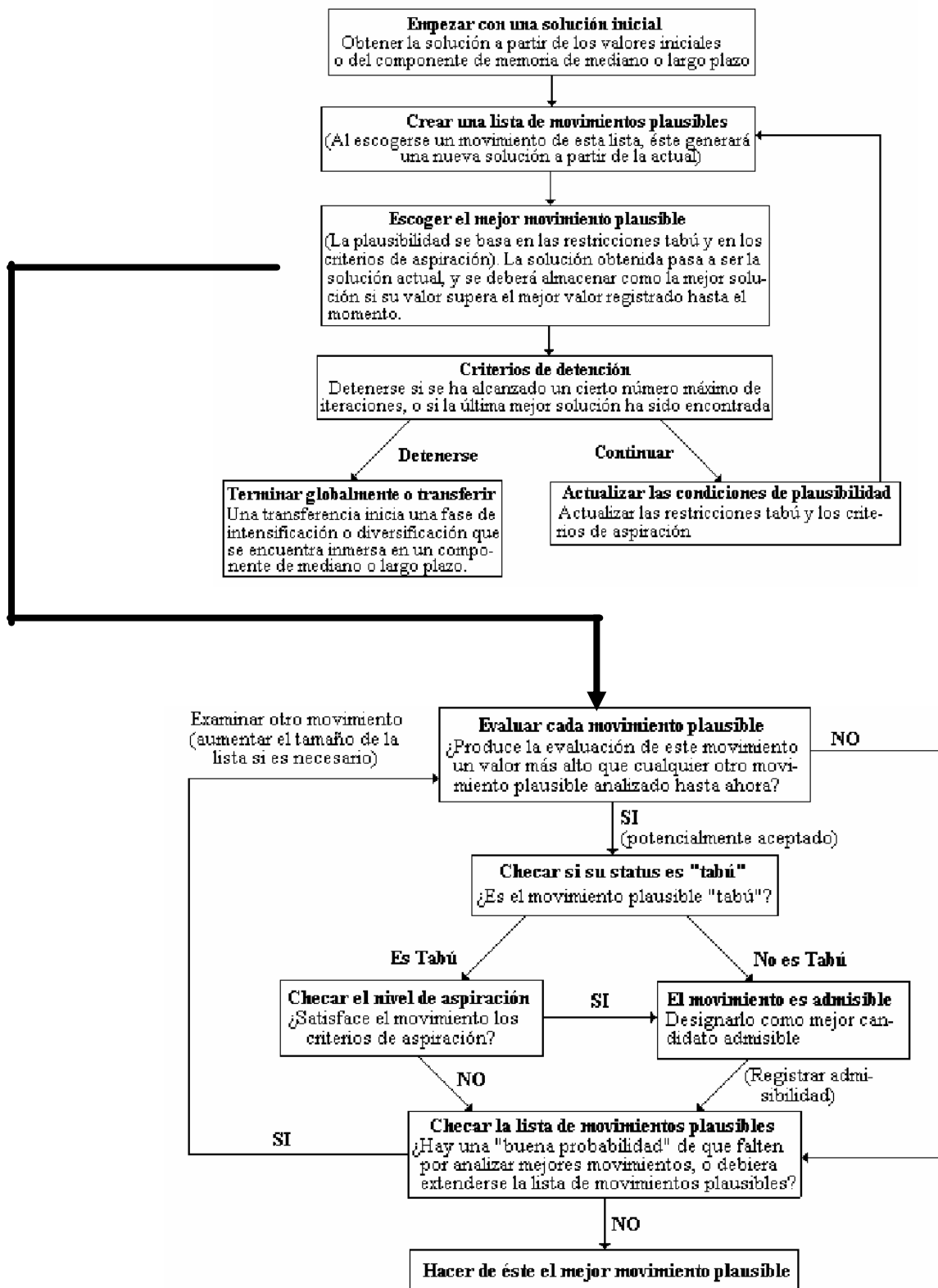


Figura 4.3 Diagrama de flujo del algoritmo basado en la “búsqueda tabú”⁴⁹

⁴⁹ Laguna, Manuel. (1994), A guide to implementing Tabu Search, “Technical Report, Graduate School of Business, University of Colorado”, Boulder Colorado, pag 78.

Particularmente para la solución del TSP, ha habido investigación a través del uso de la búsqueda Tabú, propuesto por Fred Glover en 1986 ⁵⁰. Para el caso particular del TSP, un método de búsqueda local sencilla que normalmente ha empleado Tabu Search, es el llamado “k2-opt”. Este procedimiento consiste en eliminar del tour un par de arcos que no sean adyacentes, y reemplazarlas con el único par de arcos con el cual se puede formar nuevamente el tour de manera completa. Este procedimiento se ilustra a continuación en la Figura 4.4.

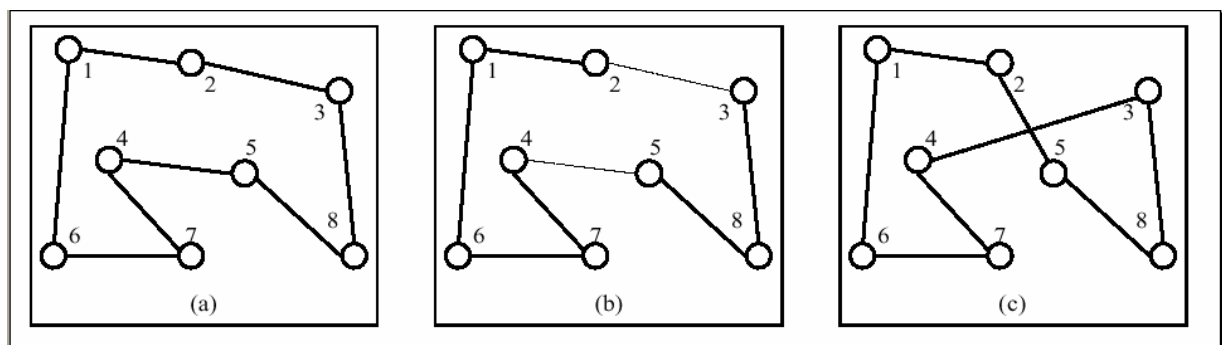


Figura 4.4 (a) Solución inicial; (b) Eliminación de dos arcos: (2,3) y (5,4);
(c) Nuevo tour sustituyendo con los arcos (2,5) y (3,4)

La meta-heurística Tabú, una vez que converge en un óptimo local, de manera alternativa, puede guiar el procedimiento de búsqueda para continuar más allá de los óptimos locales, esto quiere decir, que al no poder seguir mejorando la solución, se permite tomar otra solución aún cuando el valor no mejore, sino inclusive se degrade, esto último con la finalidad de intentar salir del óptimo local encontrado. No obstante, también existe el riesgo de caer en un ciclo, en el cual se mejora y luego se empeora la solución. Para evitar esto último, se emplea una estrategia que modifica las vecindades a ser exploradas a medida que la búsqueda avanza.

Para lograr que el algoritmo Tabú amplíe las vecindades a ser exploradas, se hace valer de unas estructuras de memoria para determinar esta vecindad modificada y así “recordar” cuales vecindades ya fueron exploradas y evitar volver a aplicar recursos computacionales en su procesamiento innecesariamente. Por ejemplo en el caso del TSP, a partir de una solución particular, una vez suprimido un par de arcos dentro del tour, estos dos arcos no pueden volver a formar parte del tour por

⁵⁰ Glover, Fred. (1986), Future paths for integer programming and links to artificial intelligence, “Computers and Operations Research”, EUA, pag 533-549.

un determinado número de iteraciones, este número de iteraciones se conoce como la permanencia tabú y es parte componente de la parametrización del algoritmo.

Análogamente, cuando un par de arcos se insertan en un tour, no podrán ser suprimidos durante un cierto número de iteraciones. Si la permanencia tabú se elige de manera adecuada, la búsqueda podrá continuar más allá de los óptimos locales sin caer en ciclos de mejoramiento y empeoramiento y así eventualmente alcanzar, si no el óptimo global del problema, sí soluciones que estén cerca de él. Es precisamente esta propiedad para evitar caer en óptimos locales, lo que ha ocasionado que “Tabu Search” tenga relevancia en el ámbito de las “meta-heurísticas” que han sido aplicadas para la solución del TSP.

Por la misma razón anterior, el postulante consideró que era importante mencionar los fundamentos del “Tabu Search” ya que bien podría haber sido otro método propuesto para atender el proyecto de investigación referido a nuestro problema de ruteo de distribución logístico.

4.7 Revisión del método basado en programación dinámica para la solución del TSP.

La programación dinámica es una técnica matemática de optimización que adolece de un problema denominado "dimensionalidad". El asunto de la "dimensionalidad" tiene que ver con la capacidad que un algoritmo tiene para mantenerse estable aún en situaciones en donde los espacios de búsqueda son muy grandes. Richard Bellman desarrolló en los años 50s las ideas básicas de la programación dinámica, postulando el principio de optimalidad el cual afirma:

*"Una política óptima tiene la propiedad de que cualquiera que sean su estado y decisión iniciales, las decisiones subsecuentes deben constituir una política óptima con respecto al estado resultante de la decisión inicial."*⁵¹

Matemáticamente el principio de optimalidad en la programación dinámica se puede expresarse como:

$$f_n(S_n) = \max_{d_n} [R_n(S_n, d_n) + f_{n-1}(S_{n-1})]$$

n =número de etapas subsecuentes en el proceso

S_n =variable de entrada a la n -ésima etapa

d_n =variable de decisión en la n -ésima etapa

$f_n(S_n)$ =retorno máximo de un proceso con n etapas y entradas S_n en la n -ésima etapa

$r_n=R_n(S_n, d_n)$ =función de retorno de la etapa n con entrada S_n y variable de decisión d_n

S_{n-1} =salida de la etapa n y entrada a la etapa $n-1$

$f_{n-1}(S_{n-1})$ =función de retorno máximo desde las etapas 1 a la $n-1$

Esta ecuación puede interpretarse de la siguiente manera: cada componente de una estructura en serie influye en todas las componentes que le siguen, y como sólo el último componente es independiente, entonces puede ser suboptimizado de manera independientemente en cada etapa. Este proceso se continúa hasta que todo el problema haya sido optimizado. Luego entonces, el número de variables de entrada en cada etapa pueden llegar a incrementarse tanto que las limitantes computacionales se vuelven sumamente serias debido naturalmente a la explosión combinatoria que ya antes hemos venido hablando. A este problema Bellman lo bautizó como la "maldición de la dimensionalidad".

⁵¹ Bellman, Richard. (1957), Dynamic Programming, Princeton University Press, Princeton Nueva Jersey, pag 83.

4.8 Revisión de métodos basados en ramificación y corte (Branch-Cut) para la solución del TSP.

La técnica de ramificación y corte fue mencionada en el apartado 4.4 como un método para resolver problemas combinatorios en términos de la solución exacta. A continuación se enumeran algunas de las investigaciones relacionadas con estrategias basadas en métodos de ramificación y corte (branch & cut) que actualmente se han desarrollado para los problemas combinatorios del tipo TSP.

1. MIP-CUT: la identificación y formulación de los cortes mínimos necesarios sigue siendo parte del estado del arte. Esta estrategia tiene la finalidad de lograr incluir las restricciones necesarias para eliminar los subtours que van quedando a partir del planteamiento del problema de asignación. Como referencias significativas tenemos a Jünger, Rinaldi and Thienel⁵².
2. CONCORDE: El algoritmo fue desarrollado por Applegate, Bixby, Chvátal y Cook sobre una plataforma en código de programación ANSI C. Incluye una fase de heurística de preproceso y algunos algoritmos generales para solución de redes. Las principales heurísticas del algoritmo de pre-proceso son la Heurística de Lin-Kernighan, Heurísticas del tipo k-opt y Búsqueda Miope (Greedy) sobre la Vecindad más próxima⁵³.
3. TSP1: Desarrollado por Volgenant y Van Den Hout en código de programación Turbo Pascal. El Algoritmo implementa estrategias de árbol "1-tree". Incluye además una heurística mejorada basada en el algoritmo general de Christofides con algunas restricciones de ruteo tipo "3-opt"⁵⁴.
4. TSP*: Desarrollado por Lee y basado en el algoritmo general de Christofides. Incluye algoritmos para la generación de cortes mínimos. El algoritmo está desarrollado en el ambiente de programación matemática AMPL⁵⁵.

⁵² Jünger, M; Rinaldi, G; Thienel, S. (2000), Practical performance of efficient minimum cut algorithms, "Algorithmica No. 26", Dinamarca, pag 172 -195. Disponible en http://www.informatik.uni-koeln.de/ls_juenger/projects/mincut.html.

⁵³ Applegate, D; Bixby, R; Chvátal, V; (1998), On the solution of traveling salesman problems. "Documenta Mathematica Extra Volume ICM III", EUA, pag 645-656. Disponible en <http://www.mathprinceton.edu/tsp/concorde.html>.

⁵⁴ Jonker, R; Volgenant, T. (1982), A branch and bound algorithm for the symmetric traveling salesman problem, "European Journal of Operational Research, No. 2", Dinamarca, pag 83-89. En <http://www.mathematik.uni-kl.de/ORSEP/contents.html>.

⁵⁵ Lee, R. (1976), Worst-case analysis of a new heuristic for the travelling salesman problem, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh PA. Disponible en <http://www.ms.uky.edu/~jlee/jlsup/jlsup.html>

5. COMBINATORICA: Desarrollado por Skiena en el ambiente de desarrollo de programación del paquete "Mathematica". Básicamente se utiliza para la generación de las fronteras inferior y superior del espacio solución ⁵⁶.
6. ABACUS: Existen dos desarrollos, uno en el ambiente de programación "CPLEX" y el otro en "XPRESS". Se trata de un algoritmo basado en estrategias enumerativas y con código para la generación de cortes mínimos así como para algoritmos para generación incremental de columnas (variables)⁵⁷.
7. MINTO: Desarrollo en el ambiente de programación "CPLEX". El algoritmo está fundamentado en la generación de cortes mínimos y estrategias de ramificación a nivel de cada nodo del árbol de búsqueda ⁵⁸.

Es importante precisar que la exposición detallada de dicha técnica será desarrollada más adelante en el apartado 8.2. La razón de lo anterior, es debido a que la aplicación de dicha técnica será de vital importancia para el proyecto de investigación en lo que concierne al diseño experimental.

⁵⁶ Skiena, S. (1990), *Implementing Discrete Mathematics: Combinatorics and Graph Theory in Mathematica*, Addison-Wesley, Redwood CA. Disponible en <ftp://ftp.cs.sunysb.edu/pub/Combinatorica/>

⁵⁷ Disponible en <http://www.oreas.de>

⁵⁸ Disponible en http://www.isye.gatech.edu/faculty/Martin_Savelsbergh/software/

4.9 Comentarios finales: resumen esquemático y otras variantes del TSP.

A continuación en la Tabla 4.2 se muestran esquemáticamente los métodos propuestos para la solución del TSP que han sido discutidos en los apartados 4.4 al 4.8.

Tabla 4.2 Resumen esquemático de los métodos propuestos para la solución del TSP.

CLASE	Método	Estrategia de Solución	Implementación	Investigadores	Referencia Bibliográfica
Heurísticas de Aproximación			TSP*: lenguaje de modelación matemático AMPL	N. Christofides	Worst-case analysis of a new heuristic for the travelling salesman problem. Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA, 1976. http://www.ms.uky.edu/~jlee/jlsup/jlsup.html
Solución Exacta	Mixed Integer Programming	Sub-tour elimination constraints	CDT: desarrollado en FORTRAN 77	Carpaneto, Dell'Amico and Toth	Solution of large-scale asymmetric traveling salesman problems. ACM Transactions on Mathematical Software, 21:394-409, 1995. http://www.acm.org/calgo/contents/
Solución Exacta y Heurísticas		El Algoritmo implementa la estrategia "1-tree". Incluye además una heurística mejorada basada en el algoritmo general de Christofides con algunas restricciones de ruteo tipo "3-opt".	TSP1: desarrollado en Turbo Pascal	Volgenant and van den Hout	A branch and bound algorithm for the symmetric traveling salesman problem based on the 1-tree relaxation. European Journal of Operational Research, 9:83-89, 1982. http://www.mathematik.uni-kl.de/~wwwi/WWWI/ORSEP/contents.html
Solución Exacta	Mixed Integer Programming	Generación de Cortes (Hiperplanos)	MIP-CUT	Jünger, Rinaldi and Thienel	Practical performance of efficient minimum cut algorithms. Algorithmica, 26:172-195, 2000. http://www.informatik.uni-koeln.de/~juenger/projects/mincut.html
Heurísticas de Aproximación	Fase de heurística de preproceso y algoritmos generales para solución de redes.	Heurística de Lin-Kernighan; Heurísticas del tipo k-opt (para k = 2,3); Búsqueda Greedy; Vecino mas próximo.	Concorde: codificado en C++	Applegate, Bixby, Chvátal y Cook	On the solution of TSP. Documenta Mathematica, Extra Volume ICM III:645-656, 1998. http://www.math.princeton.edu/tsp/concorde.html
Solución Exacta	Mixed Integer Programming	Algoritmo basado en estrategias enumerativas y con código para la generación de cortes mínimos así como para agregar columnas (variables) incrementales.	MINTO: codificado en CPLEX / OSL / XPRESS	Martin Savelsbergh	Savelsberg, M. (1995), Local search in Routing Problem with Time Windows, Annals of Operations Research, Rotherdam Holanda. http://www.isye.gatech.edu/faculty/
Heurísticas de Aproximación	Las heurísticas están basadas en métodos de inserción 2-opt y 3-opt respectivamente.		Desarrollado en Turbo Pascal	Syslo, Deo and Kowalik	Discrete Optimization Algorithms with Pascal Programs. Prentice-Hall, Englewood Cliffs, 1983. http://www.mathematik.uni-kl.de/~wwwi/WWWI/ORSEP/contents.html
Solución Exacta	Basado en algoritmos de Programación Dinámica		DYNOPT: Desarrollado en ANSI C	Balas y Simonetti	INFORMS Journal on Computing, 13:56-75, 2001. http://www.andrew.cmu.edu/~neils/tsp/
Heurísticas de Aproximación	Heurística general de Lin-Kernighan		LKH: desarrollado en ANSI C	S. Lin y B.W. Kernighan	An effective implementation of the lin-kernighan traveling salesman heuristic. European Journal of Operational Research, 126:106-130, 2000.
Metaheurísticas de aproximación	Algoritmo basado en métodos de búsqueda local dirigida y GRASP		GLTSP: codificado en C++ code	C. Voudouris y E. Tsang	Guided local search and its application to the travelling salesman problem. European Journal of Operational Research, 113:469-499, 1999. http://www.labs.bt.com/people/voudouc/downloads.htm
Metaheurísticas de aproximación	La propuesta combina el uso de Algoritmos Genéticos y de técnicas de Búsqueda Tabu (Búsqueda Tabu)		TSPGA: codificado en ANSI C code	A. Frick	An evolution program for the symmetric traveling salesman problem. In H.J. Zimmermann, editor, EUFIT'98 - 6th European Congress on Intelligent Techniques and Soft Computing, Mainz Verlag, Aachen, 1998. http://www.rz.uni-karlsruhe.de/~ul63
Metaheurísticas de aproximación	Implementación basada en Sistemas de Colonización de Hormigas (Ant Colony Systems)		OR2: codificado en el lenguaje de modelación Mathematica code	No disponible	Disponible en: http://www.softas.de/products.html
Metaheurísticas de aproximación	Implementación basada en Algoritmos de Recocido Simulado (Simulated Annealing)		RA-TSP: codificado en CPLEX 6.0 code	V. Mak y N. Boland	Heuristic approaches to asymmetric travelling salesman problems with replenishment arcs. International Transactions in Operational Research, 7:431-447, 2000. http://www.ms.unimelb.edu.au/~vmak
Solución Exacta	Mixed Integer Programming	Algoritmos orientados a la solución de problemas del tipo TSP con restricciones de precedencia especialmente las variantes relacionadas con restricciones de Ventanas de Horario (Time Window). (TSP/TW)	ASCHEUER'S Codes	Ascheuer, N; Jünger, M; Reinelt, G.	A branch & cut algorithm for the asymmetric Traveling Salesman Problem with precedence constraints, "Computational Optimization and Applications 17(1)", 2000, EUA. http://www.zib.de/ascheuer
Solución Exacta	Mixed Integer Programming	Algoritmo basado en las estrategias generales del método "Branch-and-cut" y programación MIP. Fuertemente orientado a la solución de problemas del tipo VRP (Vehicle routing problems)	SYMPHONY: codificado en ANSI C code	Ralphs D.	Disponible en http://www.branchandcut.org/
Solución Exacta	Constraint Programming (CP)	Método basado en técnicas de explotación restricciones aplicado a problemas de optimización discreta. Mayormente aplicado en problemas del TSP en donde se involucran restricciones de contigüidad.	Desarrollados codificados en C++ code	Focacci, Lodi, Milano	Disponible en http://www.or.deis.unibo.it/research_pages/Orcodes

El proyecto de investigación se sitúa en la aplicación del TSP pero para un problema de logística de distribución de mercancía a los clientes. Para este problema existen diversas variantes, una de éstas corresponde a la necesidad de algunas empresas para ejecutar dicha distribución sobre la base de esquemas de tiempos o ventanas de servicio específicos de acuerdo a las necesidades de cada uno de los clientes. Adicionalmente, un esquema común a encontrar es que la empresa disponga de un almacén central, en el cual se concentran los productos a distribuir. A partir de este almacén central se disponen unidades de transporte que se encargan de visitar a los clientes para hacer la entrega de la mercancía.

Como se puede comprender, es de esperar que las unidades de transporte sean un recurso limitado. De no ser así, pues entonces la forma en la que se podría efectuar el ruteo y la entrega de las mercancías en el menor tiempo posible, sería sobre el criterio de enviar un equipo de transporte de manera individual a cada uno de los clientes. Por tanto, lo más realista sería pensar que no se tienen tantos equipos de transporte como clientes, ya que esto resultaría sumamente costoso. Si acaso vamos al otro extremo, el caso ideal sería si la empresa dispusiera de un solo equipo de transporte para la distribución completa hacia todo el universo de clientes. En este último caso, el problema de determinar la ruta que debe de seguir el vehículo para entregar en el menor tiempo (o en la menor distancia) toda la mercancía es precisamente el TSP en su versión primordial.

No hay duda de que el escenario anterior contribuiría a que el costo fijo se reduzca drásticamente. No obstante, en la práctica puede volverse inoperante sobre la base de restricciones de tiempo y de capacidades en la distribución física. Por tanto aquí hay dos problemas en los que se tiene que pensar. En primer lugar, puede ser que el tiempo mínimo que resulte a partir de una ruta única resulte demasiado largo, lo cual en el mejor de los casos, solo ocasionaría que a algunos clientes les sea entregada la mercancía en un tiempo muy lejos de lo que ellos buscarían tener como nivel de servicio.

Por otro lado, los equipos de transporte pueden tener una cierta capacidad de carga, y por tanto sería entonces necesario que se ocupen varios equipos para poder ejecutar la distribución física de toda la mercancía que deba de ser entregada.

Puede entonces verificarse que este problema contiene dentro de sí otras consideraciones adicionales:

1. Cuántas rutas se requieren para tener un tamaño ideal de la flota de equipo de transporte.
2. Cuáles son los clientes que deben ser asignados a cada una de las rutas para hacer la entrega física.
- 3.Cuál es la ruta que debe de seguir cada uno de los equipos con la finalidad de ejecutar el reparto en el menor tiempo o distancia posible.

A lo ya antes dicho, es relevante agregar que estos tres problemas antes descritos no son independientes, sino que la solución de uno determina la del otro. Este problema se conoce como el problema de ruteo de vehículos o también denominado (VRP, Vehicle Routing Problem). A manera de describir un ejemplo, la Figura 4.5 muestra un ruteo factible para una instancia del VRP con una central de abasto, ocho puntos de entrega y tres unidades de distribución. En dicha figura, se puede verificar que cada una de las tres unidades de distribución corresponde a un sub-tour cerrado.

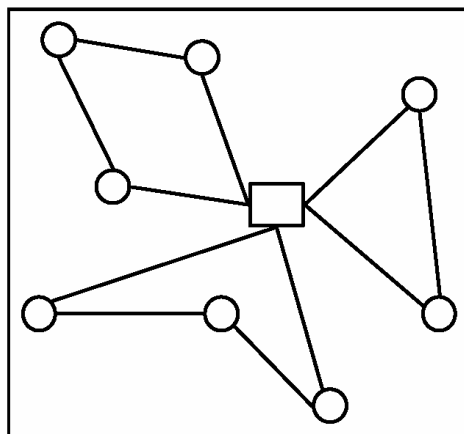


Figura 4.5 Ejemplo de un ruteo factible en un VRP
(una central de abasto, ocho clientes y tres unidades de distribución)

Finalizamos resaltando el objetivo que el postulante buscó cubrir en el presente capítulo. El problema del TSP así como nuestro problema de investigación de ruteo logístico, son problemas “NP-Hard” para los cuales las meta-heurísticas previsiblemente ofrecen buenas estrategias de solución. Las referencias expuestas en el tema de “Tabu Search” así lo exponen.

En el siguiente capítulo 5 será desarrollado el fundamento de la meta-heurística basada en los Algoritmos Genéticos. Más adelante en el capítulo 6 se hará el planteamiento formal del problema de investigación así como la exposición de los trabajos de investigación previos para el problema en

cuestión. Será en ese capítulo 6 en el cual resultará notable la ausencia relativa de la aplicación de un Algoritmo Genético como estrategia de solución para el problema planteado. La motivación del postulante para seleccionar esta meta-heurística como propuesta de solución versa precisamente sobre esta ausencia relativa que podrá ser revisada en el capítulo 6 en donde se expone la revisión bibliográfica.

Aunque hasta este momento, el uso de una meta-heurística como propuesta de solución parece ser una estrategia conveniente, es importante revisar las ventajas y desventajas de su aplicación. Por tal motivo en el apartado 5.13 del siguiente capítulo se abordará dicho tema.

Como se puede apreciar, existen varios tipos de características y propiedades que pueden hacer más fácil o difícil la solución matemática y computacional para un problema de ruteo logístico en cuestión. Lo anterior está íntimamente relacionado a la naturaleza NP-Hard de los problemas de ruteo en general. Por el motivo anterior, es importante iniciar ya el capítulo 5 en el cual se hará la exposición acerca de cual será nuestra propuesta de solución para el problema de ruteo de distribución logístico.

No hay posibilidad de que un descubrimiento se debatan solo a la casualidad. Necesariamente, la invención y el descubrimiento, ya sea en las matemáticas o en cualquier otro campo, tienen lugar a través de la combinación de las ideas.

J. Hadamard

5. Fundamentos del algoritmo genético como meta-heurística propuesta.

5.1 Introducción: breve historia del antecedente de los algoritmos genéticos.

La naturaleza ha sido siempre fuente de inspiración para el hombre. Comprender los complejos mecanismos por los cuales los seres vivos hemos alcanzado nuestra forma actual ha sido una de las tareas más ambiciosas del hombre, y sin lugar a dudas el trabajo realizado hace 140 años por el naturalista inglés Charles Darwin constituye una de las contribuciones más importantes en esta dirección.

El 1 de julio de 1858, Darwin presentó su controversial "Teoría de la Evolución Natural" ante la Sociedad Linnean de Londres, revolucionando las ciencias biológicas y cambiando incluso la filosofía del pensamiento de los seres humanos. Actualmente el paradigma neo-darwinista de la evolución sostiene que el mecanismo evolutivo de las especies e individuos está sustentado en cuatro procesos principales: reproducción, mutación, competencia y selección, todos frecuentemente resumidos en la frase "sobrevivencia del más apto y fuerte".

Alan Turing y Stanislaw Ulam se cuentan entre los científicos más célebres que pensaron en la evolución natural como el mecanismo que hizo posible el desarrollo de cualidades altamente complejas de las especies. Turing⁵⁹, argumentó que “una conexión obvia entre aprendizaje y evolución” debe de existir en los mecanismos de la cognición humana. En el Laboratorio de Los Alamos, Ulam y Von Neumann, modelaron en una computadora la velocidad con la cual las mutaciones favorables se esparcen entre los individuos de una población sujeta a los mecanismos de “sobrevivencia del más fuerte”, concluyendo que “la reproducción sexual acelera tremendamente el ritmo de crecimiento de las mutaciones favorables, comparada con la reproducción asexual que progresa de manera lineal”.⁶⁰

⁵⁹ Turing, A. (1950), Computer Machinery and Intelligence, MIND, EUA, pag 433-462.

⁶⁰ Bednarek, A. (1984), Analogies between Analogies: The mathematical reports of S.M. Ulam and his Los Alamos collaborators, University of California Press, EUA.

En el ámbito Europeo (Alemania), probablemente uno de los pioneros más importantes de la computación evolutiva sea Hans Bremmerman. Su trabajo estuvo dirigido al estudio y uso de los principios de la evolución natural como mecanismos de optimización. Bremmerman no sólo utilizó los conceptos de “aptitud”, “selección”, “mutación”, “población”, y “genotipo”, que muy pocos asociaban en aquella época con la resolución de problemas de optimización, sino que además afirmó que “la evolución biológica es un proceso de optimización”⁶¹ .

Este enunciado que parece tan simple, es la filosofía que da soporte a los métodos que en conjunto se conocen ahora con el nombre de Algoritmos Genéticos. Así pues, resulta fácil comprender que si acaso hemos de considerar a la evolución natural como un proceso de optimización, dado que la evolución ha sido capaz de optimizar organismos hasta hacerlos aptos para sobrevivir, entonces, de modelarse este principio adecuadamente, podemos emplear el mismo principio para encontrar la mejor solución para un problema de optimización matemático.

5.2 La versión americana y la versión alemana del algoritmo genético.

⁶¹ Bremermann, H. (1962), *Optimization through Evolution and Recombination*, Spartan Books, Alemania, pag 35-80.

Así pues, el concepto del Algoritmo Genético, se desarrolla a ambos lados del océano Atlántico a partir de la década de los 60's. El algoritmo correspondiente a la variante Alemana, con la intención de diferenciarlo hay quienes lo conocen como "Algoritmo basado en Estrategias Evolutivas" mientras que a la versión Americana simplemente se le denomina "Algoritmo Genético". A continuación en la Figura 5.1, se muestra un breve resumen de acontecimientos que tanto en Norte América así como en Alemania se suscitaron a lo largo de estos últimos 40 años de progreso en el desarrollo de los Algoritmos Genéticos:

ALEMANIA		NORTE AMÉRICA
1962		John Holland publica un artículo sobre Sistemas Adaptativos en el Journal del ACM
1963	Los estudiantes Ingo Rechenberg y Hans-Paul Schwefel comienzan a trabajar juntos en el túnel de viento de la Universidad Tecnológica de Berlín.	
1973	Rechenberg publica su disertación que aborda la teoría (1+1) de las EEs y la mayor parte de la nomenclatura adoptada posteriormente por los investigadores en esta área; se describe la "regla de 1/5".	
1975	La disertación de Schwefel incluye modelos y experimentos para poblaciones multi-miembro (con más de dos individuos).	Holland publica libro sobre Sistemas Adaptativos que marca el principio del actual auge de los Algoritmos Genéticos
1977	Schwefel introduce los operadores de recombinación; aparece la primera edición de su libro sobre Estrategias Evolutivas	
1985		La Primera Conferencia Internacional sobre Algoritmos Genéticos se realiza en la Universidad Carnegie-Mellon
1989		David Goldberg publica su libro sobre Algoritmos Genéticos propiciando mayor interés en el tema
1995	Schwefel publica una revisión de su libro sobre EEs . Thomas Bäck introduce las EEs a una audiencia aún mayor en 1996 con un nuevo libro de computación evolutiva	

Figura 5.1 Acontecimientos históricos en el desarrollo de los algoritmos genéticos ⁶².

A continuación en la Figura 5.2, se muestran ambas variantes del Algoritmo Genético, tanto la versión

⁶² Coello, Carlos. (1995), Introducción a los Algoritmos Genéticos, "Tecnologías de Información y Estrategias de Negocios, Año 3, No. 17", México, pag 4.

Alemana a la izquierda como versión desarrollada en los Estados Unidos a la derecha. Como puede observarse en el diagrama lógico los operadores involucrados en el algoritmo genético son básicamente los mismos, lo único que cambia es el orden en que se llevan a cabo.

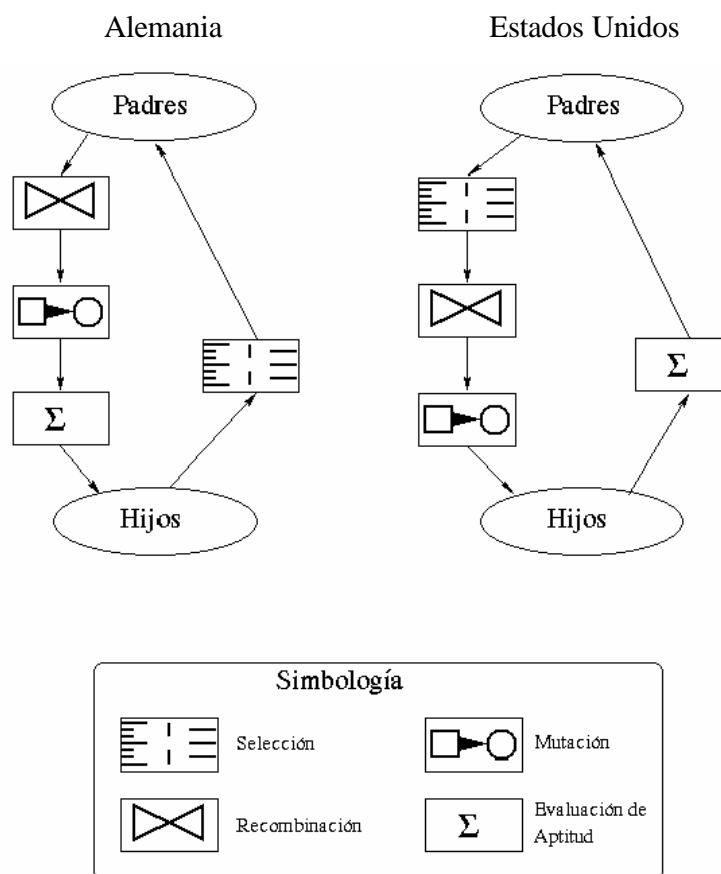


Figura 5.2 Variantes del algoritmo genético ⁶³.

Aunque existen similitudes entre ambas versiones, vale la pena diferenciar que el Algoritmo Alemán basado en las Estrategias Evolutivas se especializa en resolver problemas de optimización de funciones reales (variables continuas), mientras que los Algoritmos Genéticos se concentran mucho más en la solución de problemas de optimización discreta. Otra diferencia notable es que para la implementación computacional de los operadores de selección, cruce y mutación en los Algoritmos Genéticos, el componente estocástico es mucho más intenso que en la versión alemana debido al grado de estructuración determinístico que suelen tener estos últimos.

Aunque originalmente la versión Alemana se diferenciaba de la versión Americana en que en la

⁶³ Hoffmeister, F. (1992), Genetic Algorithms and Evolution Strategies: Similarities and Differences, "Technical Report No.1", University of Dortmund, Alemania, pag 70-120.

primera no solo se evoluciona la solución sino también los parámetros del algoritmo que propiamente llevan a converger en la solución; esta característica ha sido en los últimos años retomada por el Algoritmo Genético Americano con la finalidad de incorporarlo como parte de su estrategia en su implementación computacional.

Otra diferencia entre ambas versiones radica en que la versión Alemana, por enfocarse en la solución de problemas de optimización para variables continuas, ha dedicado mucho mas esfuerzo en lo relacionado a depurar la estrategia de guía (brújula) para mejorar su eficiencia durante el proceso de búsqueda del espacio solución en la fase de escalamiento. Para entender lo anterior, basta preguntarse: ¿qué debemos seleccionar, un organismo cerca de la solución pero mal orientado, o uno alejado de la solución pero correctamente orientado?

En la opinión del postulante, la principal diferencia entre la versión Alemana y la versión Americana estriba en que en la primera los organismos no solo están constituidos por su propio material genético (genotipo), sino que también poseen su propio control (fenotipo). Sería algo así como permitirle a cada organismo tener su propia personalidad y voluntad. De esto último entonces, resulta razonable preguntarse si esta doble característica no resulta excesiva para tan solo implementar el mecanismo de selección natural en los organismos de la población. Es precisamente este problema de doble constitución de la versión alemana lo que representa su talón de Aquiles en la práctica. Debemos puntualizar que en este asunto en la versión Americana, el ajuste de los parámetros se caracteriza no al nivel de cada organismo sino a nivel global de la población.

De cualquier manera es claro comprender que la versión Americana nos es más útil por enfocarse hacia problemas de optimización discreta que es exactamente hacia donde nos estamos dirigiendo, ya que nuestro problema de investigación está referido a un caso de ruteo de distribución en el área de logística.

5.3 Introducción a los parámetros de operación del algoritmo genético.

El Dr. John Holland, investigador de la Universidad de Michigan, consciente de la importancia que la selección natural ha tenido en el proceso de la evolución natural, a fines de la década de los 60s desarrolló una técnica que permitió simular dicho proceso biológico a través de un algoritmo computacional. Su objetivo original era lograr que las computadoras aprendieran por sí mismas ⁶⁴. Eventualmente la técnica que inventó Holland se llamó "Algoritmo Genético" tras la publicación de su libro en 1975. Los Algoritmos Genéticos son técnicas de búsqueda que permiten explorar de forma eficiente el espacio solución de un problema. El proceso de búsqueda de la solución óptima está basado en la teoría de la evolución de Darwin mediante el cual, los mecanismos de genética y de selección natural aplicados en el proceso evolutivo estimulan la formación de nuevos organismos mejor adaptados al medio ambiente.

Otra definición de lo que es un Algoritmo Genético se ofrece por John Koza (traducción libre):

“Se trata de un algoritmo matemático paralelo que transforma un conjunto de objetos matemáticos individuales con respecto al tiempo usando operaciones modeladas de acuerdo al principio Darwiniano de reproducción y supervivencia del más apto, y tras haberse presentado de forma natural una serie de operaciones genéticas entre las que destaca la recombinación sexual. Cada uno de estos objetos matemáticos suele ser una cadena de caracteres (letras o números) de longitud fija que se ajusta al modelo de las cadenas de cromosomas, y se les asocia con una cierta función matemática que refleja su aptitud.” ⁶⁵

En los últimos años, la comunidad científica internacional ha mostrado un creciente interés en esta nueva técnica la cual se basa en los mecanismos de selección que utiliza la naturaleza, de acuerdo a los cuales los individuos más aptos de una población son los que sobreviven al adaptarse más fácilmente a los cambios que se producen en su entorno. Los Algoritmos Genéticos gozan hoy en día de enorme popularidad en todo el mundo, debido a su generalidad y sencillez conceptual.

La aplicación más común de los algoritmos genéticos ha sido para la solución de problemas de optimización matemática, en donde han mostrado ser eficientes. Sin embargo, no todos los problemas

⁶⁴ Holland, John. (1975), *Adaptation in Natural and Artificial Systems*; University of Michigan Press, EUA, pag 211.

⁶⁵ Koza, John. (1992), *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, The MIT Press, EUA, pag 319.

pueden ser apropiados para esta técnica. A continuación se enumeran algunas características que el problema debe tener para aplicar un algoritmo genético de acuerdo a la empírica del postulante:

1. El espacio de búsqueda para el problema debe estar delimitado dentro de ciertos límites: Lo recomendable es aplicar el algoritmo para resolver problemas que tengan espacios de búsqueda discretos. Esta característica le ofrece naturalidad al algoritmo y le permite atender inclusive espacios de solución inclusive grandes. Su aplicación al tratarse de espacios de búsqueda continuos resulta menos eficiente por lo cual preferentemente se limita a casos donde exista un rango de soluciones relativamente pequeño.
2. Es necesario definir una función de aptitud que nos indique qué tan buena o mala es cada característica revisada en el organismo. La función de aptitud no es otra cosa que la función objetivo del problema de optimización matemático. La función de aptitud debe ser capaz de "castigar" a las malas soluciones (organismos), y de "premiar" a las buenas, de forma que sean estas últimas las que se propaguen con mayor rapidez en la población.
3. Las soluciones para el problema deben ser codificadas de una forma que resulte fácil su implementación algorítmica. La codificación más común es a través de cadenas binarias, aunque también se han empleado otras codificaciones basadas en números reales y letras. La codificación binaria ha gozado de mucha popularidad debido a que es la técnica que propuso originalmente Holland, y además porque resulta muy sencilla de implementar.

Existen diversos operadores que se utilizan en los algoritmos genéticos para guiar el proceso evolutivo y lograr la generación de nuevas poblaciones de individuos a partir de los existentes. A continuación se describen los operadores que el postulante implementará en el algoritmo genético que será aplicado como propuesta de solución para el problema de investigación:

1. Población y su función de desempeño: se parte de la existencia de una población inicial, que dará paso a nuevas generaciones a través de la ejecución de un proceso similar a lo que ocurre en la naturaleza.

2. Cromosomas y genes: son soluciones potenciales, que van formando la población de organismos. Los cromosomas se constituyen a su vez por genes los cuales contienen la información genética codificada del organismo.
3. Selección: competencia para realizar el cruzamiento de organismos.
4. Cruzamiento o Reproducción: generación de nuevos organismos a partir de la combinación del contenido genético de dos organismos antecesores que ya existan en la población.
5. Mutación: alteración del contenido genético de nuevos organismos sin que intervengan los padres, a partir de pequeños cambios en uno o más genes de un determinado cromosoma.

Aunque todos los parámetros antes expuestos son importantes, en la Figura 5.3, se resaltan dos de ellos de manera particular: la selección natural y el cruce o reproducción.

Fundamento de un AG.

Selección Natural y Reproducción Sexual

1. El 1er proceso determina cuales organismos tienen mayor oportunidad de sobrevivir.
2. El 2do asegura que estos organismos tengan mayor oportunidad de reproducirse y propagar sus genes.

La finalidad es:

Simular el proceso de selección natural para probar el desempeño de cada organismo en competencia.

Figura 5.3 Fundamento del algoritmo genético.

En los siguientes puntos trataremos de manera específica cada uno de estos parámetros para su mejor comprensión.

5.4 La población de organismos y la función de desempeño.

Los componentes de la población son denominados organismos o individuos. La estructura de los organismos es determinada a priori y es la misma para toda la población. La estructura precisa de los organismos es dependiente del dominio del problema, lo cual implica que para cada problema tiene que idearse una representación adecuada del organismo (genoma). No obstante lo anterior, la operación del Algoritmo Genético es independiente del dominio del problema.

El tamaño de la Población “P”, representa ser uno de los parámetros más importantes del Algoritmo Genético, esto es la representación del conjunto de soluciones potenciales para el problema. El tamaño de la población puede variar a lo largo de las generaciones del Algoritmo, aunque usualmente dicho parámetro permanece estático.

Una característica que los problemas deben de tener, es la definición de una medida comparativa de las soluciones (organismos) que compiten. Debe de existir un mecanismo derivado del dominio del problema que nos permita asignar una medida de desempeño (o fitness) a cada uno de los organismos de la población y que éste realmente sea representativo de su calidad como solución. A este parámetro se le denomina medida o desempeño (o también aptitud). Un organismo con mayor aptitud representa una mejor solución para el problema, lo cual en las condiciones específicas de éste puede representar una solución correcta o inclusive la misma solución óptima para el problema.

Es evidente que lograr una alta correlación entre el valor de aptitud y la cercanía a la solución que cada individuo representa resulta finalmente crítico. Esto se propicia mediante la representación o codificación apropiada del problema, y por supuesto, mediante una función de aptitud que correctamente logre hacer este mapeo asignando un valor de calidad adecuado a cada organismo de acuerdo a su desempeño. Precisamente esto es lo que permite indicar qué tan buena es una solución con respecto al resto de la población.

5.5 La definición del cromosoma del organismo y sus genes.

Las unidades básicas de codificación que permiten distinguir a cada uno de los atributos de un ser vivo son lo que conforman los genes de un individuo. Son estos mismos genes los que permiten que los atributos más deseables se transmitan a sus descendientes al momento del acto de reproducción

La principal característica de un Algoritmo Genético es que éste no requiere de información de dominio o conocimiento respecto al procedimiento para solucionar un problema. La justificación para el uso de representaciones binarias como esquema principal de codificación de los organismos en los Algoritmos Genéticos se remonta al trabajo desarrollado por Holland en 1974. La codificación binaria permite un mayor grado de paralelismo implícito en el algoritmo debido simplemente a la cantidad combinatoria de bloques esquemáticos que una representación binaria puede llegar a cubrir respecto a una representación por ejemplo del tipo decimal.

Un bloque esquemático o simplemente esquema, no es otra cosa que un patrón que se visualiza al revisar en varios organismos de la población, un subconjunto de genes que comparten similitudes a lo largo de la longitud cromosómica del genoma ⁶⁶. Es luego fácil comprender que al haber una mayor diversidad de esquemas a poder estar siendo permutados, entonces esto incrementa la diversidad y por consiguiente la probabilidad de que un buen bloque pueda llegar a generarse que devenga a su vez en la solución óptima para el problema.

Lo anterior, no significa que no puedan utilizarse codificaciones de mayor cardinalidad, es decir que sean diferentes a la codificación binaria. Lo único es estar consciente de que la codificación binaria finalmente será aquella que podrá ofrecer una mayor cantidad de esquemas posibles que cualquier otro mecanismo de codificación. Por tanto, entre mayor cantidad de genes esté representando a un genoma, mucho más sensibilidad tendrá el algoritmo para la formación de bloques constructores. Más adelante se dedicará un apartado para revisar más a detalle el tema de la codificación de los organismos (revisar el apartado 5.8).

⁶⁶ Goldberg, D. (1995), Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley Pub Co, University of Massachusetts, EUA, pag 45-87.

En la Figura 5.4, se ejemplifica aún más el concepto de codificación en el desarrollo de una representación binaria para el manejo de las soluciones en el Algoritmo Genético. En la figura se menciona como es que dentro del algoritmo genético es posible incorporar en el genoma de los organismos, ciertas propiedades y características que a priori se “estima”, por parte del modelador, puedan ser favorables para la evolución de la población. A este componente del algoritmo genético, el postulante le nombra como el efecto de “La mano de Dios”.

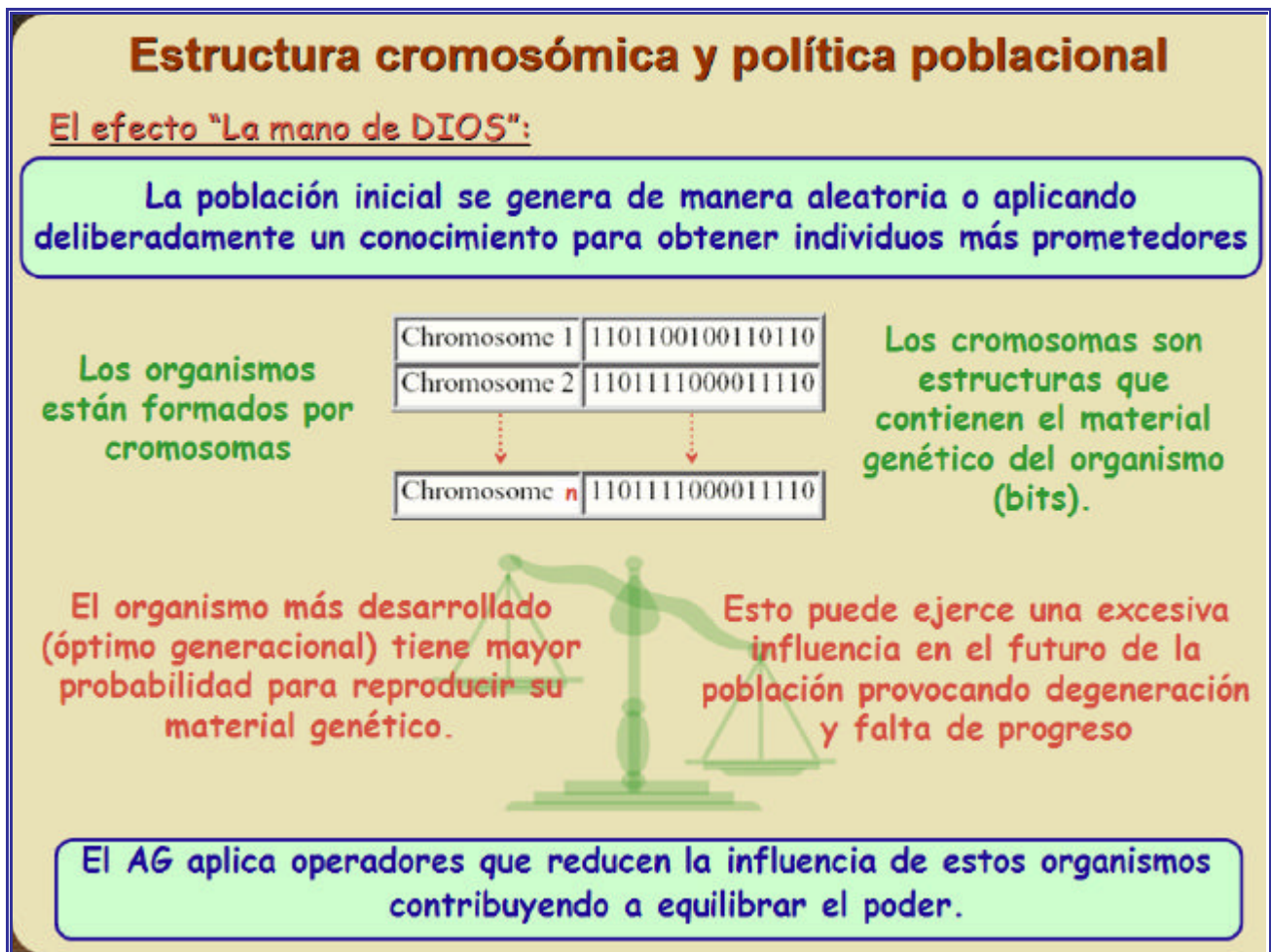


Figura 5.4 Desarrollo de la estructura cromosómica y política poblacional.

5.6 El operador de selección.

La selección es el operador que escoge preferentemente a los organismos con mayor aptitud de una población. El grado en el cual mejores valores de aptitud son preferidos sobre los más pobres se denomina presión de la selección. Existen dos métodos de selección que fueron aplicados en la implementación del Algoritmo Genético propuesto para determinar cuales de los organismos de la población son los que tendrán mayor probabilidad de descendencia: el de Ruleta y el de Torneo.

1. La Ruleta: fue desarrollado por Goldberg⁶⁷, el método consiste en crear una ruleta en la que cada organismo tiene asignada una fracción del área de la ruleta en proporción a su grado de aptitud. Sin que nos refiramos a una función de aptitud en particular, supongamos que se tiene una población de 5 cromosomas cuyas aptitudes están dadas por los valores mostrados en la siguiente Tabla 5.1:

Tabla 5.1 Ejemplificación del método de selección de la ruleta

Cromosoma No.	Cadena	Aptitud	% del Total
1	11010110	254	24.5
2	10100111	47	4.5
3	00110110	457	44.1
4	01110010	194	18.7
5	11110010	85	8.2
Total		1037	100.0

Con los porcentajes mostrados en la cuarta columna de la Tabla 5.1, podemos pasar ahora a construir la ruleta de la Figura 5.5. Esta ruleta se gira una cantidad “n” de veces para determinar qué organismos se seleccionarán. La cantidad “n” esta determinada al menos por la cantidad de organismos que se desea mantener en la población. Debido a que a los organismos más aptos se les asignó un área mayor de la ruleta, se espera que sean seleccionados más veces que los menos aptos.

⁶⁷ Goldberg, David. (1989), Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Publishing Company, EUA, pag 412.

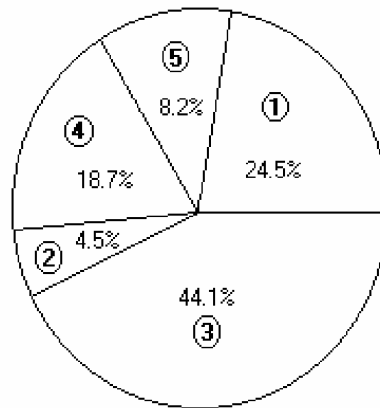


Figura 5.5 Ruleta que representa los valores de aptitud de la tabla 5.1

2. El Torneo: este método se basa en el sorteo de la población para luego hacer competir a los organismos que la integran en grupos de tamaño predefinido (normalmente en parejas). La finalidad de éste método es simular un torneo en el cual resultan ganadores aquéllos organismos que tengan valores de aptitud más altos. Esta técnica garantiza la obtención de múltiples copias de los mejores individuos de entre los progenitores de cada siguiente generación.

A continuación en la Figura 5.6, se explica de manera breve la forma como está íntimamente relacionada la función de desempeño de un organismo con respecto a la forma como actúa el operador de selección en el algoritmo genético. Es decir, entre mayor sea la función de desempeño de un organismo mayor probabilidad tendrá en ser seleccionado para el proceso de cruzamiento sexual que se explica en el siguiente apartado.

Operador de Selección

El material genético en los organismos determina la función de desempeño.

La función de desempeño:

Simula la presión adaptativa que ejerce el medio ambiente sobre los organismos determinando que tan bien resuelve un organismo un problema.

Org.	f(x) Desempeño	P(x) Prob.
01001	5	19%
10000	12	46%
01110	9	35%

Los organismos son ordenados y seleccionados de acuerdo a su función de desempeño $f(x)$.

Entre mejor sea el organismo, mayor probabilidad de descendencia tendrá para transmitir su material genético.

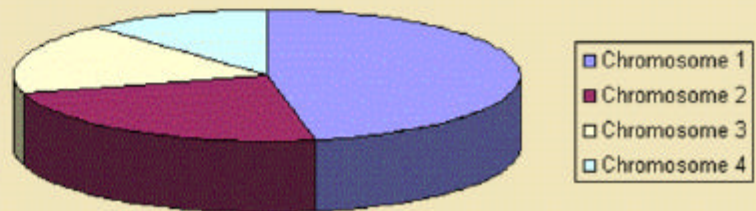


Figura 5.6 Funcionamiento del operador de selección genético.

5.7 El operador de cruce o reproducción.

Una vez realizada la selección, se procede a la reproducción sexual o cruce de los organismos seleccionados. En esta etapa, los sobrevivientes intercambian material genético y sus descendientes forman la población de la siguiente generación. El operador de cruce (o reproducción) es aquel por el cual los organismos de una población intercambian información genética. La implementación computacional del algoritmo genético propuesto, considera dos formas de reproducción sexual:

1. Uso de un punto único de cruce: éste se escoge de forma aleatoria sobre la longitud de la cadena que representa el cromosoma, y a partir de él se realiza el intercambio de material de los 2 individuos, tal y como se muestra en la Figura 5.7.

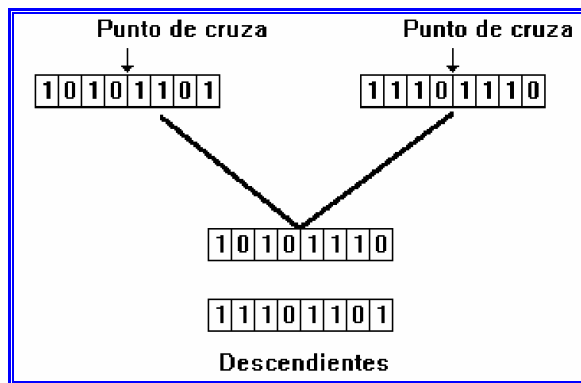


Figura 5.7 Un solo punto de cruce. Cada pareja de organismos da origen a 2 descendientes.

2. Uso de dos puntos de cruce: cuando se usan 2 puntos de cruce, se procede de manera similar, pero en este caso el intercambio se realiza en la forma mostrada en la Figura 5.8. Con este método de cruce entre 2 organismos se mantienen los genes de los extremos, y se intercambian los del centro.

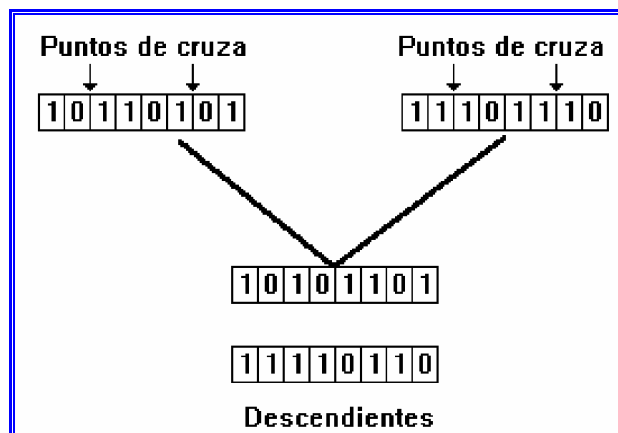


Figura 5.8 Dos puntos de cruce.

Normalmente la cruce se maneja dentro de la implementación del algoritmo genético como un parámetro expresado como un porcentaje que indica con qué frecuencia se efectúa dicho evento. Esto significa que no todas las parejas de organismos se cruzarán, sino que habrá algunas que pasarán intactas a la siguiente generación. Existe una variante del Algoritmo Genético en la que el organismo más apto a lo largo de las distintas generaciones no se cruza con nadie, y se mantiene intacto hasta que surge otro individuo mejor que él, que lo desplazará. A esta variante del algoritmo genético se le denomina “Elitismo” en virtud precisamente de lograr mantener el genoma “elite” de generación en generación hasta que aparezca un organismo con mejor desempeño que lo reemplace.

El operador de cruce se maneja como un parámetro en el Algoritmo Genético mediante un porcentaje que indica con qué frecuencia se efectuará dicho evento. El porcentaje de cruce normalmente es de alrededor del 60%, mientras que el porcentaje de mutación que a continuación se va a exponer en el siguiente apartado normalmente nunca supera el 5% ⁶⁸. En general, el porcentaje de mutación se fija mucho más bajo que el cruce.

A continuación en la figura 5.9, se explica de manera esquemática el mecanismo mediante el cual el proceso de cruzamiento ocurre. Es interesante verificar el comentario que se hace en función a como puede llegar a reducirse la diversidad genética del nuevo organismo al manipularse el material genético proveniente de un solo organismo padre y no de dos.

⁶⁸ Coello, Carlos. (1995), Introducción a los Algoritmos Genéticos, “Tecnologías de Información y Estrategias de Negocios, Año 3, No. 17”, México, pag 5-11.

Cruce Genético y Reproducción

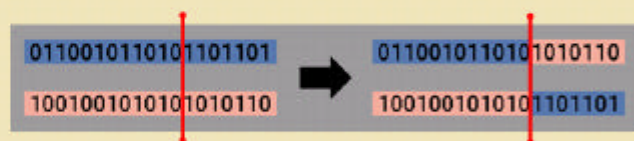
Este es el proceso más **CRUCIAL** de la evolución:

A diferencia de la clonación donde se reduce la diversidad genética ya que se manipula un solo material genético.

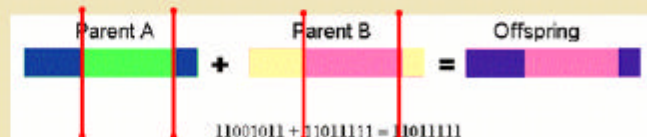
La idea básica es:

Intercambiar el contenido genético de dos individuos promedio con la expectativa de poder obtener un organismo mejor.

Operador de cruce simple



Operador de cruce doble



La probabilidad para la reproducción sexual de los organismos es proporcional a su nivel de desempeño.

Figura 5.9 Proceso de cruce genético y reproducción

5.8 El operador de mutación.

Además del operador de selección y de cruce, existe otro operador llamado mutación. La mutación es el operador que causa cambios aleatorios en los organismos de la población. El cambio se realiza a uno de los genes del cromosoma elegido aleatoriamente. Cuando se usa una representación binaria, el gene seleccionado se sustituye por su complemento, es decir un cero se cambia por uno y viceversa. Este operador permite la introducción de nuevo material cromosómico en la población, tal y como sucede en los equivalentes sistemas biológicos.

El operador de mutación aplicado en el Algoritmo Genético no es otra cosa más que un procedimiento de búsqueda escalando la colina (Hill climbing). Este proceso de escalamiento en el Algoritmo Genético es esencialmente aleatorio para producir las alteraciones esperadas en el contenido genético de los organismos. Las mutaciones que resulten en mejoras de desempeño en los organismos se conservan y el resto se desechan. No obstante, los desplazamientos a lo largo de los ejes del sistema de coordenadas durante el proceso de escalamiento, es poco probable que sean en línea recta sobre la dirección de búsqueda más corta hacia la solución óptima global. Esto significa que la trayectoria de búsqueda que sigue el proceso de mutación sigue un camino en forma de zigzag más o menos perpendicular al vector correspondiente a la dirección de la solución óptima. Este proceso de búsqueda en zigzag antes descrito se puede ejemplificar en la Figura 5.10.

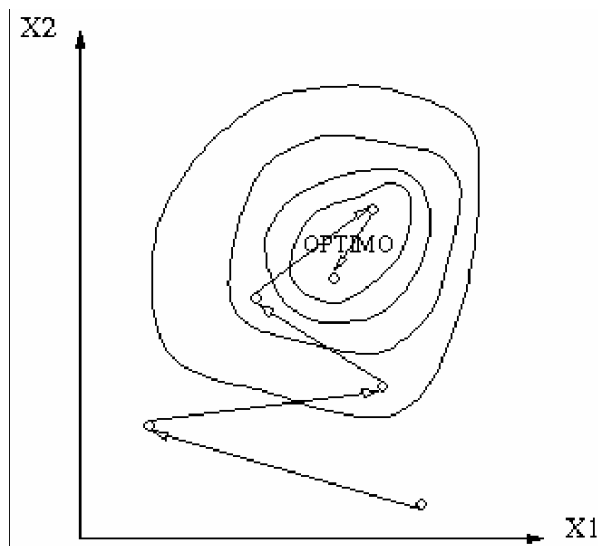


Figura 5.10 Proceso de escalamiento para la búsqueda de la solución óptima.

A continuación en la Figura 5.11, se desarrollan breves explicaciones respecto al factor de mutación y de elitismo ya antes mencionados. Es importante precisar que aunque los dos factores “parezcan” opuestos, en realidad lo que ocurre es que el primero se concentra en diversificar el contenido del genoma dentro de una población y el segundo busca perpetuar de generación en generación aquel genoma que resulte con mejor desempeño.

Factor de Mutación y Elitismo


Agrega “Sangre Fresca” a la población para:

Mantener la diversidad genética y evitar la degeneración poblacional

Operador de mutación simple 0110010110101101101 ➡ 0110110110101101101

La mutación contribuye a ampliar nuevas configuraciones genéticas:

Incrementa la probabilidad para evolucionar hacia un genoma óptimo.



NO obstante, dicho factor NO debe ser alto ya que esto convertiría la evolución en un proceso completamente casual.

El elitismo contribuye a evitar perder organismos excepcionales:

MODULA el reemplazo generacional de los padres por los hijos. Solo los mejores organismos, sean padres o hijos, trascienden.

La “inmortalidad” de un organismo existe mientras siga siendo competitivo, haciendo frente al control poblacional.

Figura 5.11 Afectación del factor de mutación y elitismo en el algoritmo genético.

5.9 Funcionamiento general del algoritmo genético simple.

La operación del algoritmo genético en su versión más simple puede ilustrarse con el siguiente pseudo-código presentado en la Figura 5.12:

```
generar población inicial, G(0);
evaluar G(0);
t:=0;
repetir
    t:=t+1;
    generar G(t) usando G(t-1);
    evaluar G(t);
hasta encontrar una solución;
```

Figura 5.12 Pseudo-código del algoritmo genético simple ⁶⁹.

Con la finalidad de ofrecer una breve explicación al pseudo código anterior, iniciamos primero al generar aleatoriamente la población inicial, la cual estará constituida por un conjunto de organismos, o cadenas de caracteres que representan las soluciones posibles del problema. A cada uno de los organismos de esta población se le aplicará la función de aptitud a fin de saber qué tan buena es la solución que está codificando en el interior de su contenido genético. Habiendo evaluado la aptitud de cada organismo, se procede a la selección de los que se cruzarán en la siguiente generación. En este sentido es presumible pensar, que el Algoritmo Genético escogerá a los "mejores" organismos.

Un aspecto interesante en un Algoritmo Genético es determinar su punto de terminación. Si acaso supiéramos la respuesta óptima a un problema, entonces la respuesta a esto último es trivial ya que sólo es necesario incluir en el Algoritmo Genético una indicación que haga concluir el procedimiento de cómputo una vez que el algoritmo converge en la solución previamente definida como óptima. Sin embargo, eso casi nunca es posible, o por lo menos no es lo que se supone sea lo deseable. Debido a lo anterior, normalmente se usan 2 criterios principales de detención:

1. Iterar el Algoritmo Genético hasta un número máximo de generaciones.
2. Detenerlo una vez la población se haya estabilizado, es decir, cuando todos o la mayoría de los individuos tengan la misma aptitud. A esto se le llama homeóstasis.

⁶⁹ Buckles, Bill. (1992), Genetic Algorithms , IEEE Computer Society Press, EUA, pag 109.

En términos prácticos al aplicar el segundo método, una característica que se presenta cuando la población se estabiliza, es que esto guarda una relación estrecha respecto a la probabilidad de que exista la generación de mejores organismos. Dicho de otro modo, al haber organismos muy parecidos entre sí, la probabilidad de un enriquecimiento en la solución disminuye debido precisamente a esa falta de variedad genética de donde echar mano para mejorar la solución.

Con la finalidad de representar gráficamente lo antes expuesto, en la Figura 5.13, se ofrece una mayor explicación respecto al funcionamiento básico del algoritmo genético.



Figura 5.13 Funcionamiento del algoritmo genético.

5.10 El problema de la codificación de la solución (genoma) en los algoritmos genéticos.

En el asunto de la codificación en los Algoritmos Genéticos, es necesario diferenciar entre lo que es un genotipo y un fenotipo. El genotipo es precisamente la cadena cromosómica del organismo, la cual es utilizada para contener su información genética. A diferencia tenemos el fenotipo, el cual corresponde a los valores que se calculan una vez que se aplica el proceso de decodificación sobre el genotipo. Ambos elementos están íntimamente relacionados ya que un buen Algoritmo Genético requiere de un fenotipo apropiado que pueda contribuir a medir el nivel de desempeño de los organismos y a su vez un buen fenotipo requiere de un genotipo apropiado que logre “maximizar” la probabilidad de generar bloques esquemáticos que permitan evolucionar el desempeño del fenotipo⁷⁰.

Así pues, el paralelismo implícito al que se refería Holland en su Algoritmo Genético es precisamente el hecho de que mientras el algoritmo determina las mejores aptitudes de los organismos de una población, al mismo tiempo determina en forma implícita las características promedio de un número obviamente mucho mayor de cadenas cromosómicas. Lo anterior se logra a través precisamente de la explotación de los bloques constructores que se van verificando como patrones característicos en los organismos de la población.

No obstante la ventaja de la codificación binaria, es importante precisar que también existen dos desventajas principales:

1. En los problemas de optimización continua, es común que se lleguen a requerir cadenas cromosómicas demasiado largas lo cual propicia un desempeño extremadamente lento en la eficiencia del algoritmo al buscar la solución.
2. La aplicación de los operadores de cruce naturales pueden llegar a producir soluciones ilegales sobre la base de una codificación binaria.
3. La codificación binaria en forma natural no logra mapear adecuadamente el espacio de búsqueda con el espacio de representación. Dicho de una manera sencilla, quiere decir que aunque dos soluciones sean contiguas en el espacio solución, en la representación codificada de ambos, éstos pueden llegar a ser tan diferentes por varios órdenes de magnitud (muchos bits).

⁷⁰ Hollstein, R. (1991), Artificial Genetic adaptation in computer control systems, Phd Thesis, University of Michigan, EUA, pag 84.

En lo referente al último punto expuesto se han desarrollado investigaciones que han venido a resolver dicha dicotomía entre la adyacencia del espacio de búsqueda respecto a la adyacencia en el espacio de representación (o codificación). Uno de los métodos recientemente propuestos (1998), ha sido el código de Gray ⁷¹. La virtud de la codificación en Gray es en principio que continua siendo binaria, solo que se le añade un procedimiento de transformación a través de la operación booleana “XOR” aplicada cada uno de los bits contenidos en el cromosoma de derecha a izquierda.

Investigaciones han comprobado que la aplicación de este tipo de codificaciones, han resultado en una mejora sustancial en el desempeño del Algoritmo Genético ya que se disminuye la cantidad de óptimos locales como consecuencia del uso de una representación mucho más apegada al propio espacio de búsqueda ⁷².

En lo referente a la codificación de números reales que requieren el manejo de decimales con alta precisión, el problema de la codificación binaria es inevitablemente crítico, con o sin el uso de la codificación de Gray mencionado antes. Para esto se han desarrollado representaciones basadas en el estándar de la IEEE, mediante el cual este tipo de números siguen siendo codificados en forma binaria pero no a partir de una representación en formato decimal natural sino a partir de una representación en formato exponencial.

Este tipo de pre-codificación del número real permite con tal solo 32 bits representar cualquier número real de precisión simple. De los 32 bits, los primeros 8 se aplican para la representación del exponente y el resto de los otros 24 bits se utilizan para otorgar precisión en la mantisa. Ante el beneficio que otorga el seguir manteniendo una representación binaria el costo de lo anterior es el proceso de decodificación ya que el cromosoma deberá ser interpretado de su representación en binario a su representación exponencial y luego finalmente a su valor en formato decimal natural.

⁷¹ Whitley, D. (1998), Representation issues in Neighborhood Search and Evolutionary Algorithms, John Wiley and Sons, University of Sussex, England, Capítulo 3, pag 39-57.

⁷² Mathias, K. (1994), Transforming the search space with gray coding, Proceedings of the IEEE International Conference on Evolution Computation, Piscataway New Jersey, EUA, pag 213-218.

5.11 Trabajos previos de investigación: aplicación de los algoritmos genéticos para optimización en problemas de logística.

Es interesante comprobar que el historial de la aplicación de los Algoritmos Genéticos en el campo de los problemas de logística es realmente incipiente en términos relativos. La revisión bibliográfica realizada nos lleva a identificar que la cantidad de publicaciones hasta el año 2002 registradas como aplicación total de los Algoritmos Genéticos es de 16,524 documentos. De igual modo de estas publicaciones tan solo 584 hacen referencia a su aplicación en la Logística, es decir únicamente el 3.5% del total. A continuación en la Figura 5.14 se evidencia dicha situación en la cual se utiliza una escala logarítmica en la que se muestra la cantidad de publicaciones año con año.

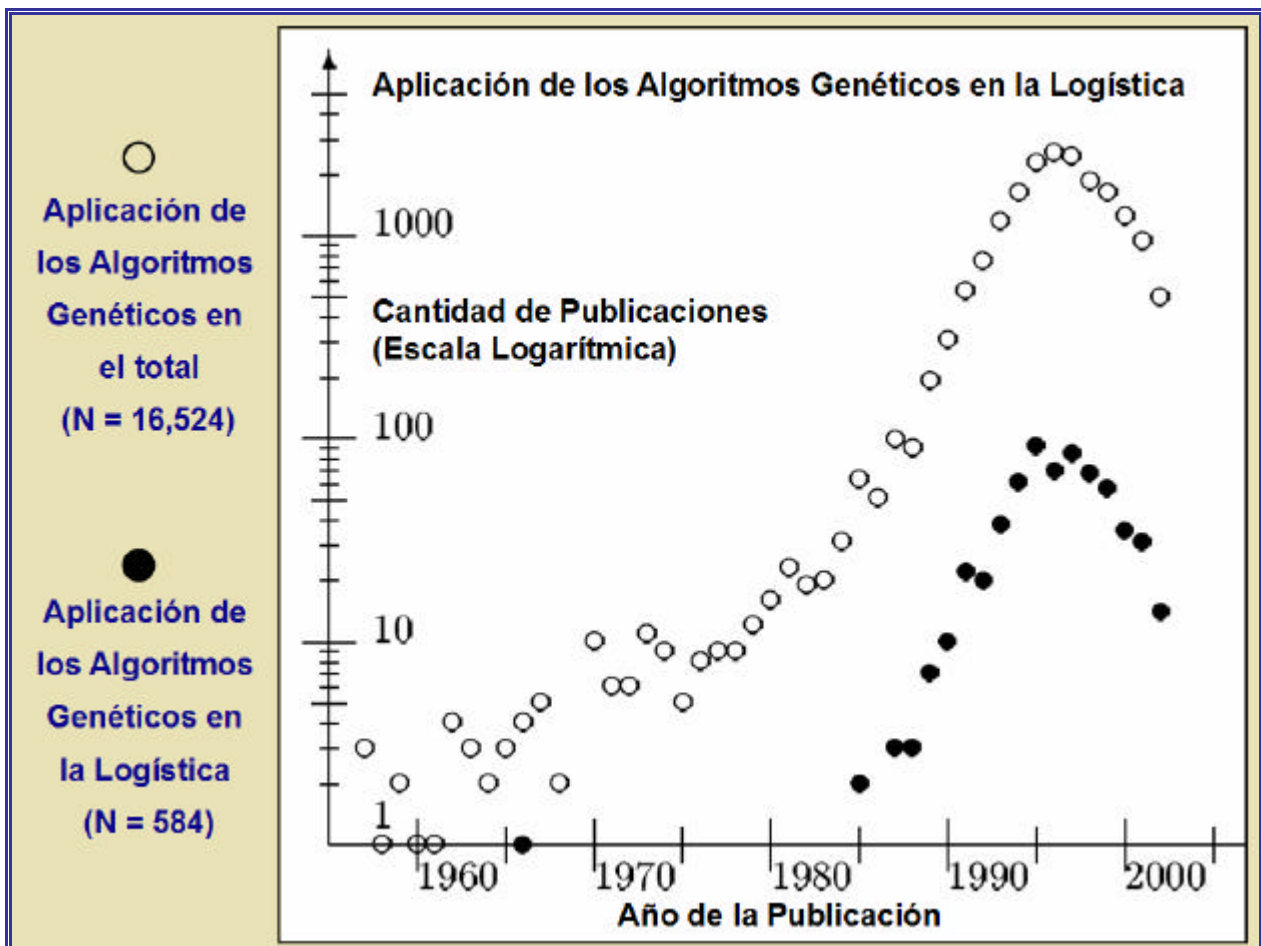


Figura 5.14 Relación entre la cantidad de aplicaciones de los Algoritmos Genéticos en el campo de la Logística con respecto al total⁷³.

⁷³ Alander, Jarmo. (2003), An Indexed Bibliography of Genetic Algorithms in Logistics, Department of Engineering, University of Vaasa, Finland, pag 11.

Como complemento a la figura anterior, podemos estratificar la cantidad de publicaciones que han sido desarrolladas de este mismo universo. En la tabla 5.2 se muestra la estadística por año.

Tabla 5.2 Cantidad de publicaciones hechas por año de Algoritmos Genéticos en la Logística ⁷⁴.

AÑO	FREC
1966-1986	3
1987	3
1988	3
1989	7
1990	10
1991	22
1992	20
1993	38
1994	61
1995	92
1996	69
1997	85
1998	67
1999	57
2000	35
2001	31
2002	14
TOTAL	617

En la tabla 5.3 se muestra la estadística por país de origen en la cual las primeras 2 columnas hacen referencia al total de las aplicaciones y las últimas 2 columnas a las aplicaciones de los algoritmos genéticos únicamente en el campo de la logística.

Tabla 5.3 Publicaciones por país de los Algoritmos Genéticos en la Logística respecto al total ⁷⁵.

PAIS	TOTAL		LOGISTICA	
	FREC	%	FREC	%
Estados Unidos	4.829	29%	155	27%
Japón	2.320	14%	62	11%
Gran Bretaña	1.838	11%	54	9%
Alemania	1.266	8%	44	8%
Finlandia	534	3%	30	5%
China	803	5%	26	4%
Francia	441	3%	23	4%
Italia	505	3%	23	4%
Corea del Sur	408	2%	21	4%
Canada	262	2%	15	3%
Australia	422	3%	14	2%
Taiwan	363	2%	14	2%
España	276	2%	13	2%
Grecia	102	1%	8	1%
India	264	2%	8	1%
Isreal	110	1%	7	1%
Singapur	136	1%	7	1%
Holanda	181	1%	7	1%
Brazil	134	1%	6	1%
Croacia	26	0%	6	1%
Otros	1.304	8%	41	7%
TOTAL	16.524	100%	584	100%

⁷⁴ Alander, Jarmo. (2003), Idem , pag 11.

⁷⁵ Alander, Jarmo. (2003), Idem , pag 12.

Concentrando ahora la atención en las aplicaciones de los Algoritmos Genéticos en el campo de la Logística, podemos ahora clasificar su frecuencia en función al tema particular. A continuación en la Tabla 5.4, se pueden apreciar la contribución de ha tenido los algoritmos genéticos en diversos aspectos relacionados a la función de la logística.

Tabla 5.4 Publicaciones por Tema de Investigación de los Algoritmos Genéticos en la Logística⁷⁶.

TEMA DE INVESTIGACION	FREC
Problema del TSP	175
Telecomunicaciones	88
Control y Paralelismo	71
Transportación	62
Ruteo	45
Otros Logística	28
Secuenciación	25
Estrategias Híbridas	22
Técnicas de Cruzamiento	18
Tráfico	14
Redes Neuronales	12
Lógica Confusa	9
Estrategias Evolutivas	9
Tamaño de Población	6
TOTAL	584

Finalmente en la Tabla 5.5, podemos revisar la clasificación de las publicaciones de acuerdo al tipo de documento que haya sido desarrollado.

Tabla 5.5 Publicaciones por Tema de Investigación de los Algoritmos Genéticos en la Logística⁷⁷.

TIPO DE DOCUMENTO	FREC
Artículos (Proceedings)	339
Artículos (Journals)	199
Tesis de Doctorado	19
Tesis de Maestría	6
Libros de Texto	5
TOTAL	568

Es interesante mencionar que la aplicación de los Algoritmos Genéticos en problemas reales de logística, ha tenido un desarrollo híbrido. Es decir, no han sido aplicados de manera holística, sino que han sido implementados como parte componente de una metodología más extendida en la cual se incluyen diversos algoritmos de pre-procesamiento y post-procesamiento. Existen implementaciones

⁷⁶ Alander, Jarmo. (2003), Idem , pag 10.

⁷⁷ Alander, Jarmo. (2003), Idem , pag 9.

metodológicas que en una primera fase, utilizan heurísticas de simplificación de rutas⁷⁸. También es posible el uso de procesos de reinicialización una vez detectada alguna convergencia local.

Existen metodologías en las cuales, la fase correspondiente al proceso evolutivo, genera un porcentaje de la población inicial utilizando técnicas tipo GRASP, dejando que el resto de la población se genere de forma naturalmente aleatoria. La generación de soluciones iniciales a través de técnicas GRASP, permite obtener soluciones bastante buenas con un mínimo esfuerzo computacional⁷⁹. La única desventaja de insertar en la población inicial un por ciento de soluciones de este tipo, es que esto irremediamente lleva a una convergencia local en iteraciones tempranas del algoritmo. No obstante, esta situación puede ser resuelta a través de la aplicación de operadores de reinicialización⁸⁰.

En lo referente a la técnica de codificación de la solución en el cromosoma, existen diversas formas de implementarlo. Una de estas es la representación basada en el orden, esto es, a través de listas de tamaño variable con los nodos del camino cerrado que conforman la solución. Cada solución comienza incorporando a una ruta vacía un nodo aleatorio como primer elemento y continúa añadiendo al azar el nodo V_k del subconjunto V de nodos adyacentes al último nodo incorporado. Este procedimiento finaliza cuando todos los nodos están incorporados a la ruta terminada. Sea (c_1, c_2, \dots, c_n) la ruta obtenida. Para cerrar la ruta, entonces se adiciona a la ruta el nodo (c_{n+1}, \dots, c_1) tal que $(c_n, c_{n+1}, \dots, c_1)$ obtenga la ruta de distancia mínima entre c_n y c_1 ⁸¹.

Tocante a la función de desempeño o "fitness", ésta típicamente está representada por la distancia de la ruta. Al igual, el operador de selección mayormente utilizado es el de torneo (tournament en inglés), en el cual se van identificando dos soluciones de forma aleatoria y seleccionando la mejor. Este proceso se repite dos veces con la población dada, de forma tal que la mejor solución será seleccionada dos veces mientras que la peor es eliminada⁸².

En lo que concierne a la fase de reproducción, lo más adecuado es utilizar operadores de cruzamiento

⁷⁸ Ling, Wang (1999), An effective hybrid optimization strategy for job-shop scheduling problems, Computers and Operations Research. Vol. 28, pag 585-596.

⁷⁹ Feo, T, (1995), Greedy Randomized Adaptive Search Procedures, Journal of Global Optimization, Vol. 6, pag 109-133.

⁸⁰ Gulnara, M, (2002), Un enfoque híbrido basado en metaheurísticas para la solución del cartero rural, Congreso latinoamericano de Investigación de Operaciones, Chile, pag 3-7.

⁸¹ Hakim, Al (2001), An analogue genetic algorithm for solving job shop scheduling problems, International Journal of Production Research, vol. 39, No 7, pag 1537-1548.

⁸² Coello, Carlos. (1995), La importancia de la representación en los Algoritmos Genéticos, Tecnologías de Información y Estrategias de Negocios, Año 3, No. 17, Enero de 1995, pag 5-11.

que estén personalizados específicamente para el problema en cuestión. La razón de lo anterior es debido a que un operador de cruce que no esté bien planeado, puede llegar a generar frecuentemente organismos que no satisfagan los requerimientos de una ruta para el problema⁸³. Aunque en tales casos es posible aplicar procedimientos reconstructivos a través de la incorporación de material genético a partir del último nodo que pudiera estar a salvo en el cromosoma, no obstante, lo mejor siempre será crear un operador de cruzamiento que evite generar organismos infactibles en la población. Indistintamente en el producto del cruzamiento de dos padres se preservan los dos mejores entre los dos padres y los dos hijos.

En lo que toca al operador de mutación, idealmente esté debe estar inmerso en el procedimiento para la formación de los hijos. No obstante en algunos otros casos, el operador de mutación se maneja en forma individual con la finalidad de que el algoritmo genético rápidamente converja y pueda así acceder deliberadamente a fases especializadas que estén incluidas en la metodología general⁸⁴.

Hablando de fases de post-procesamiento respecto al algoritmo genético, una estrategia de implementación puede ser en la cual una vez generada una nueva población, se busca entonces mejorar cada elemento de la población utilizando rutinas de simplificación de rutas. Estas rutinas de simplificación pertenecen a la fase de búsqueda local y pueden ser dirigidas a través de técnicas basadas en GRASP para mejorar cada ruta construida⁸⁵.

Otra estrategia de post-procesamiento al algoritmo genético puede ser implementada a partir de una población de soluciones elite obtenida durante la ejecución de la fase evolutiva. En estos casos, es posible aplicar a cada organismo de la población elite, un operador genético de inversión modificado. Este operador genético de inversión modificado solo requiere de un padre para generar a un hijo. El operador de inversión antes descrito, invierte un sub-camino de la ruta completa, la cual es formada entre dos puntos de ruptura seleccionados como parte del proceso, produciendo de esta manera una nueva solución⁸⁶. La nueva ruta formada no requiere de un proceso de verificación, ya que dicho operador preserva mucho de los enlaces entre los genes del cromosoma evitando el “rompimiento” de

⁸³ Palominos, Pedro. (2001). Influencia del Tipo de Cruza y Tamaño de la Población en la Calidad de la Solución de un Algoritmo Genético, Revista del Instituto de Investigación Operativa, Volumen 6, N°3, pag 3-8.

⁸⁴ Gruttner, E. (2002), Algoritmos Genéticos en recorridos óptimos de líneas de transporte público, Departamento de Ingeniería y Ciencias de la Computación, Universidad de Concepción, Chile, pag 8-15.

⁸⁵ Colomer, José. (1999), Los modelos de cuatro etapas: utilidad y limitaciones, Universidad Politécnica de Valencia, Valencia, España, pag 6-20.

⁸⁶ Gendreau, M. (1992), New Insertion Post-optimization Procedures for the Traveling Salesman Problem, Operations Research, Vol 40, pag 1086-1094.

la ruta. Lo anterior hace efectivo su aplicación en un proceso de mejora de soluciones, aunque exista la posibilidad de que el hijo generado no sea mejor que el padre. Así entonces, en cada iteración de la fase de post-procesamiento se va aplicando el operador de inversión modificado al 100% de la población, manteniendo siempre la mejor solución entre el padre y su propio hijo.

En lo referente a la estrategia de finalización generacional de las poblaciones, hay algunas en las cuales cuando se detecta una convergencia a un óptimo local se aplica el operador de reinicialización, generando una nueva población inicial. Lo anterior permite acceder a las diversas regiones del espacio solución a ser explorado a lo largo del algoritmo general⁸⁷.

Como es posible constatar, existe una diversidad interesante de estrategias a poder ser implementadas en un Algoritmo Genético. Para el caso de nuestro problema de investigación, es importante mencionar que como parte del proceso de inicialización, se incorpora un procedimiento particular para generar la población inicial. En términos generales nuestro procedimiento parte de un nodo designado como origen el cual entonces es ubicado como primer gen en el cromosoma. Luego a partir de una matriz de adyacencia, se busca aleatoriamente un vecino a este nodo⁸⁸. Luego entonces, este vecino se coloca en el segundo gen y así sucesivamente se repite el proceso para cada nodo hasta alcanzar el nodo destino. Este procedimiento se aplica tantas veces como cromosomas se ocupen para completar los organismos que se requieran para cubrir la población inicial.

Este procedimiento “dirigido” para la generación de la población inicial a partir de una matriz de adyacencia previamente construida, ciertamente también puede ser tomado en cuenta como estrategia para el desarrollo de un operador de cruzamiento dirigido⁸⁹. La ventaja de esta estrategia “dirigida” es que evita caer al algoritmo genético propuesto en óptimos locales. Lo anterior se logra computacionalmente hablando a través de identificar como inválidas aquellas rutas que no contengan alguna particularidad o propiedad esperada en la conformación del contenido genético. En nuestro caso, se logra implementar lo anterior a través de la aplicación de factores de penalización para todas aquellas secuencias genéticas que aparecieran en el cromosoma que se estuvieran alejando de alguna propiedad deseada en la matriz de adyacencia.

⁸⁷ Bäck, T. (1998), On The Behavior Of Evolutionary Algorithms In Dynamic Environments, Proceedings of the Fifth IEEE Conference on Evolutionary Computation, IEEE Press, Pag. 446-451.

⁸⁸ Hansen, P. (1998), Variable Neighborhood Decomposition Search, University of Montreal, pag 53-98.

⁸⁹ Mladenovic, N. (1995), A Variable Neighborhood Algorithm: a New Metaheuristic for Combinatorial Optimization, Abstract of papers presented at Optimization Days, Montreal, pag 12.

5.12 Ventajas y desventajas del algoritmo genético como estrategia de solución.

A continuación enumeramos algunas de las ventajas que se tiene al aplicar los Algoritmos Genéticos en problemas de optimización matemática:

1. No se necesitan conocimientos específicos sobre el problema que se busca resolver.
2. Paralelismo implícito: operan de forma simultánea con varias soluciones, en vez de trabajar de forma secuencial como las técnicas tradicionales.
3. Robustez: resultan menos afectados respecto a las heurísticas tradicionales para el manejo de convergencias prematuras hacia óptimos locales (soluciones sub-óptimas).
4. Son fáciles de implementar en los sistemas computacionales con arquitectura en paralelo.

En la Figura 5.15, se describe la forma mediante la cual dichas ventajas resultan ser preponderantes respecto a la solución de los problemas NP-Hard en lo general y del problema de investigación en lo particular.



Figura 5.15 Ventajas en la aplicación de los algoritmos genéticos.

En confrontación con lo anterior, a continuación se enumeran algunas de las desventajas que se tiene en su aplicación:

1. Se aplican operadores de convergencia probabilísticos, en vez de determinísticos.
2. Están fuertemente en dependencia respecto a sus parámetros de operación ya antes mencionados, tales como el tamaño de la población, número de generaciones, el porcentaje de mutación, el porcentaje de cruce, etc.
3. Dependiendo de las características propias del problema, pueden llegar a tardar mucho en converger, o quizás peor aún nunca llegar a converger en absoluto⁹⁰.

Para finalizar el presente apartado, citaremos algunos de las conclusiones obtenidas en otros proyectos de investigación previos en donde se desarrollan análisis comparativos del desempeño de los algoritmos genéticos respecto al uso de otras meta-heurísticas para la solución de problemas de secuenciación en general.

Reeves en 1995 desarrolla un análisis comparativo entre un algoritmo genético y un algoritmo basado en enfriamiento simulado. La investigación de Reeves concluye que el algoritmo genético obtiene mejores soluciones en un tiempo menor que su contraparte⁹¹. Por otro lado, Croce en su investigación concluye que el algoritmo genético obtiene mejores resultados que la meta-heurística Tabu-search con la única atenuante del tiempo computacional⁹².

Finalmente podemos citar a Cheng, el cual en su investigación se enfoca en comprobar el porqué es superior la aplicación del algoritmo genético con respecto al uso de otras meta-heurísticas. Cheng, expone en sus conclusiones que enfriamiento simulado y tabu-search desarrollan un búsqueda unidireccional a través del uso de una sola solución como candidato. En contraste, continua Cheng, el algoritmo genético desarrolla una búsqueda multi-direccional en paralelo a través del uso de múltiples soluciones potenciales. Es precisamente en ese sentido, que el algoritmo genético hace uso de los mejores organismos en la población para explorar diferentes secciones del espacio solución simultáneamente⁹³.

⁹⁰ Coello, Carlos. (1995), Introducción a los Algoritmos Genéticos , "Tecnologías de Información y Estrategias de Negocios, No. 17", México, pag 5-11.

⁹¹ Reeves, C. (1995), Genetics algorithms flow shop sequencing, Computers & Operations Research, Vol 22, EUA, pag 5-13.

⁹² Croce, F. (1995), A genetic algorithm for the job shop problem, Operations Research, Vol 22, EUA, pag 15-24.

⁹³ Cheng, R. (1997), Genetic algorithm and Engineering design, John Wiley and Sons, New York EUA, pag 7.

5.13 Variantes adicionales en los algoritmos genéticos.

Para nuestro problema de investigación, aunque ya hemos comentado que se trata de optimización discreta, la problemática de la codificación binaria sigue estando presente. En este sentido, la mayor parte de las investigaciones se han entonces venido concentrando en la necesidad de desarrollar operadores de cruce especializados que logren mantener las bondades de la codificación binaria pero que a la vez eviten generar soluciones inválidas durante el proceso de reproducción⁹⁴.

Investigaciones han propuesto el uso de cromosomas con una representación matricial y binaria la cual permite realizar operaciones de cruce y de mutación que no son posibles realizar en los Algoritmos Genéticos convencionales. Adicional a la ventaja antes descrita, el mecanismo de selección para la cruce hace que la población final no converja hacia un contenido genético similar de organismos, evitando así la posibilidad de caer en óptimos locales que redundan en una degeneración de la población. Por otra parte, el aplicar un criterio pseudo-elitista en la selección de individuos para cada nueva población, permite una convergencia en el valor de la función de adaptación del mejor individuo encontrado.

Dentro de la fase de selección de la nueva población es posible que el algoritmo emplee el principio del espacio poblacional extendido que permite que tanto padres como descendientes tengan las mismas posibilidades de competir por la supervivencia. De esta manera el algoritmo genera una nueva población de igual tamaño a la original, combinando elitismo y aleatoriedad en la formación de ella. En una primera instancia, se eligen los individuos más adaptados, el mejor de cada población (inicial, hijos y mutantes). Enseguida, para completar el número constante de individuos, se seleccionan representantes de cada población de manera aleatoria.

Una última variante del Algoritmo Genético que se va a exponer, es aquella que resulta cuando es necesario el tratar con cromosomas no de tamaño fijo como sucede en el caso natural, sino cuando el tamaño del cromosoma puede ir variando (longitud variable) de acuerdo a los diferentes estados posibles durante el proceso de búsqueda de la solución. La necesidad para manejar cromosomas de longitud variable, existe con la finalidad de capacitar al algoritmo para ir cambiando el nivel de precisión a medida que éste avanza en el proceso de búsqueda. No obstante, también hubo otra razón después, y esta proviene de la necesidad de ir agregando o disminuyendo de manera dinámica

⁹⁴ Deb, K. (1995), Simulated Binary crossover in evolutionary optimization, "Complex Systems No. 9", EUA, pag 115-148.

elementos (o variables) del cromosoma a medida que se va convergiendo hacia una solución en el espacio de búsqueda.

Así fue como nació la versión “desordenada” del Algoritmo Genético (Messy Genetic Algorithms) ⁹⁵. La idea fundamental de esta variante es la de empezar con cromosomas de longitud corta que permitan de manera gradual y al mismo tiempo relativamente más rápido identificar aquellos bloques esquemáticos que mejor características provean a los organismos de la población para de esta manera gradualmente propagar estos esquemas hacia cromosomas de mayor longitud. Así pues, la versión “desordenada” del Algoritmo Genético se desarrolla en dos fases.

La primera es la llamada fase primordial que servirá a identificar aquellos esquemas que servirán como bloques constructores a propagar a su vez en la segunda fase. La segunda fase se le denomina, fase de yuxtaposición. El gran asunto en esta variante corresponde en determinar que tan largos deben ser los esquemas a ser generados en la primera fase. La dificultad de lo anterior obedece a que si son demasiado pequeños se puede caer en el error de que no contengan suficiente material genético para el proceso de propagación y mejoramiento en la fase de yuxtaposición. Por el otro lado, si acaso son demasiado largos, pues entonces se redundaría en un Algoritmo Genético simple de solo una fase. Las investigaciones reportadas a la fecha en esta variante del Algoritmo Genético, reportan relativamente pocos casos de éxito ⁹⁶.

⁹⁵ Goldberg, D. (1996), Don't Worry, be messy , "Proceedings of the 9th International Conference on Genetic Algorithms", The MIT Press, Cambridge Mass, EUA, pag 24-30.

⁹⁶ Chowdury, M. (1997), Messy Genetic Algorithms based new learning method for fussy controllers, "Proceedings of the IEEE International Conference on Industrial Technology", The MIT Press, Shangai China, pag 110-130.

5.14 Comentarios finales.

En los 90's, una gran parte de la investigación en esta área se ha concentrado en el desarrollo de mejoras al desempeño de los algoritmos genéticos. Se han propuesto nuevas técnicas de representación, selección y cruce. Por ejemplo, el uso de los códigos de Gray y la codificación dinámica han superado algunos de los problemas asociados con la representación de valores reales mediante cadenas binarias. También se han propuesto técnicas adaptativas que varían dinámicamente los parámetros de control (porcentajes de mutación y cruce), en contraposición con el esquema estático tradicional. Otras innovaciones notables son los algoritmos genéticos distribuidos y los algoritmos genéticos paralelos⁹⁷.

No obstante el avance relativo en la aplicación de los Algoritmos Genéticos, el fundamento matemático de estos sigue siendo un área abierta de investigación. Este interés se ha reflejado recientemente en México, a través del centro de investigación LANIA (Laboratorios Nacionales de Informática Avanzada).

Con el presente capítulo damos por terminado el marco conceptual de la tesis. En el capítulo 3 definimos a la logística como área de aplicación y algunos de sus problemas más representativos desde donde establecimos de manera genérica hacia donde enfocaríamos el problema de investigación. Luego en el capítulo 4 definimos el concepto NP-Hard en relación al problema de investigación previamente establecido. En dicho capítulo se establecieron las técnicas de optimización más prominentes utilizadas para dar solución a los problemas tipo NP-Hard. Finalmente, en este capítulo 5 establecimos el marco teórico conceptual referido a la técnica de optimización basada en Algoritmos Genéticos, la cual está siendo propuesta como técnica de solución para el problema de investigación.

En el siguiente capítulo 6 nos abocaremos a definir formalmente y matemáticamente el problema de investigación. En el capítulo 7 haremos el planteamiento de los objetivos e hipótesis y en el capítulo 8 el diseño experimental.

⁹⁷ Srinivas, M. (1999), Genetic Algorithms: A Survey, IEEE Computer, EUA, pag 17-26.

6. Planteamiento del problema de investigación

6.1 Introducción al problema de investigación.

En la primera parte del documento de tesis (capítulo 1 y 2) se expusieron los antecedentes del proyecto, mediante los cuales, se introduce al lector en el tema de la logística así como en el factor educativo como elemento habilitador para la aplicación de la investigación de operaciones. La segunda parte correspondió al marco conceptual (capítulos 3 al 5). En el capítulo 3, se expusieron de manera general algunos de los problemas en la práctica de la logística. En el capítulo 4 se exponen de manera particular los fundamentos matemáticos necesarios para atender los problemas de logística de ruteo. Finalmente en el capítulo 5 se exponen los fundamentos teóricos referentes al procedimiento meta-heurístico propuesto para dar solución al tipo de problemas tratados particularmente en el capítulo 4.

A partir de ahora damos inicio a la 3era y última parte del documento de tesis el cual corresponde al desarrollo del proyecto de investigación. En el presente capítulo 6 se hará el planteamiento formal del problema de investigación. En el capítulo 7 se formularán los objetivos, justificaciones e hipótesis del proyecto. En el capítulo 8 se expondrá el diseño experimental para probar la hipótesis y finalmente en los capítulos 9 y 10 se expondrán los resultados y las conclusiones del experimento respectivamente.

Los problemas de optimización discreta se diferencian de los de optimización continua en que los primeros son aquellos que involucran variables cuyas soluciones necesariamente requieren ser del tipo “entero” debido al tipo de fenómenos que inherentemente tratan. La propiedad de integralidad de las variables involucradas no es trivial ya que esto representa el corazón mismo del problema redundando en la condición de intratabilidad de los problemas del tipo “NP-Hard”. Aunque este último aspecto esta fuera del tratamiento del proyecto de investigación, es importante precisar que la mayor parte de los tipos de problemas combinatorios redundan en esta misma característica de complejidad.

Todos estos problemas “NP-Hard” poseen como característica común el que ninguno puede ser resuelto de manera eficiente a través de algún algoritmo conocido, de existir, entonces podría propiciar

la solución para varios problemas de su mismo tipo de la misma manera⁹⁸. La condición de intratabilidad que se propaga a lo largo de todos los miembros del grupo de problemas “NP-Hard”, ha propiciado que la comunidad científica se halle enfrentada en encontrar un método (algoritmo) para hacerle frente. Nuestro problema de investigación pertenece a esta clase de problemas.

La administración del proceso de distribución físico presenta diversas variantes a ser atendidas en la toma de decisiones. Al nivel estratégico podemos referenciar decisiones cuyo objetivo es dar respuesta a la ubicación óptima de plantas y centros de distribución. En el nivel táctico podemos hablar de decisiones referentes a la determinación óptima del tamaño y configuración de la flota de distribución. El presente proyecto de investigación apunta hacia la administración del nivel operativo de la distribución el cual, como más adelante se definirá, se concentra en optimizar la distancia, tiempo y/o costo del proceso de distribución.

Aunque más adelante se establecerán las características que diferencian y justifican nuestro proyecto de investigación, resulta inevitable referenciar nuestro problema de investigación a esfuerzos científicos anteriores. La mejor similitud que se tiene respecto a investigaciones previas, se logra al referenciar nuestro proyecto al problema de entrega y recolección con ventana de horario (Pickup and delivery problem with time windows, PDP-TW). En general, la investigación en esta área del conocimiento se ha enfocado principalmente en la atención de modelos simplificados cuyo objetivo ha sido el servir como herramientas y bloques de construcción para la solución de instancias del mundo real⁹⁹.

En términos matemáticos, podemos decir que nuestro problema PDP-TW resulta ser un problema extremadamente difícil de resolver en términos del esfuerzo computacional requerido, aún para instancias pequeñas del problema. Más aún, Savelsberg demuestra que tan solo encontrar una solución factible (no necesariamente óptima) representa ser un problema difícil de solucionar (NP-Hard)¹⁰⁰. Nuestro problema PDP-TW se puede definir como la conjunción de 3 problemas de optimización combinatoria:

1. El problema de ruteo (Routing) del TSP.
2. El problema de secuenciamiento (Scheduling) del SJSS: debido a las restricciones de

⁹⁸ Parker, R; Rardin, R. (1998) Discrete Optimization, Academic Press, New York EUA, pag 67.

⁹⁹ Cook, W; Rich, Jennifer. (1999), A parallel cutting-plane algorithm for the vehicle routing problem with time windows, Computational and Applied Mathematics Rice University, Houston Texas EUA, pag 2.

¹⁰⁰ Savelsberg, M. (1995), Local search in Routing Problem with Time Windows, Annals of Operations Research, Rotherdam Holanda, pag 285-305.

precedencia y las ventanas de horario que requieren ser consideradas.

3. El problema de empacamiento (Packing): por las restricciones de entrega y recolección que requieren ser tomadas en cuenta respecto a la capacidad de carga del vehículo.

Como se puede apreciar, nuestro problema PDP-TW es una variante mucho más general que el problema básico del TSP. La razón de lo anterior es básicamente a que en el PDP-TW aparecen muchas más restricciones que tomar en consideración. Así por ejemplo partiendo del PDP-TW:

1. Si se eliminan las restricciones de entrega y de recolección, entonces nuestro PDP-TW se convierte en un problema de ruteo de vehículos con restricciones de ventana de horario (Vehicle Routing Problem with Time Windows, VRP-TW).
2. Si al VRP-TW se le eliminan las restricciones de capacidad de carga del vehículo, entonces el VRP-TW se convierte en un problema de ruteo con restricciones de ventana de horario (Traveling Salesman Problem, TSP-TW).
3. Más aún, si al TSP-TW se le eliminan las restricciones de ventanas de horario, entonces el TSP-TW se convierte en el problema básico del TSP.

Por si acaso hubiera duda de la complejidad del PDP-TW, podemos referirnos a un problema cercano como es el TSP-TW con la finalidad de buscar simplificar nuestro problema original. Tsitsiklis en 1992, investiga la variante más simplificada del TSP-TW. Esta resulta ser aquella en la que los tiempos de proceso asociados a cada uno de los nodos de la red, tienen asignado un valor igual a cero. Tsitsiklis en su investigación demuestra que para esta versión simplificada del TSP-TW aún obtener una solución apenas factible, ya no digamos la óptima, resulta ser igualmente un problema NP-Hard¹⁰¹.

En la década de los 90's, se han logrado grandes avances en el aspecto teórico para la solución del problema referido al PDP-TW. Ciertamente uno de los mayores avances en el tema ha sido la incorporación de características del mundo real tales como la consideración de las ventanas de horario así como las restricciones de precedencia que hasta antes habían sido "relajadas" debido a su complejidad matemática. Cuando hablamos de la relajación de un problema, nos referimos a omitir de manera temporal las restricciones de integralidad de las variables discretas involucradas en el problema. Más adelante en el apartado 8.2, se atenderá de manera específica este tema.

¹⁰¹ Tsitsiklis, J. (1992), Special cases of traveling salesman and repairman problems with time windows, "Networks No. 22", EUA, pag 263-282.

6.2 Planteamiento formal del problema de investigación:

Ruteo de distribución para un vehículo con entrega y recolección de producto con restricciones de ventana de horario negociables (SPDP-sTW).

El problema del PDP-TW puede ser a su vez visto en dos variantes principales. Savelsbergh en 1995, distingue el PDP-TW en su variante para un solo vehículo SPDP-TW (Ruteo de distribución para un vehículo con entrega y recolección de producto con restricciones de ventana de horario), de aquella otra variante para vehículos múltiples MPDP-TW. El primer caso se trata de un TSP-TW restrictivo mientras que el segundo se trata de un VRP-TW restrictivo. Obviamente el calificativo de restrictivo, proviene de la necesidad de considerar en el problema la operación de entrega y de recolección así como la capacidad de carga del vehículo en cuestión. Nuestro proyecto se concentra en el primer caso, es decir en el SPDP-TW. Más aún, se trata de un SPDP-sTW debido a la existencia de ventanas de horario negociables para cada cliente, donde la “s” que antecede al “TW” proviene del inglés “soft”. Así pues, hemos finalmente ubicado a nuestro problema de investigación taxonómicamente hasta su nivel más específico. Podemos ahora apuntar el siguiente planteamiento para nuestro problema. El problema de investigación será entonces planteado formalmente en términos de dos componentes. El primero será la definición del objetivo del problema y el segundo se referirá a las condiciones de operación para el problema. A continuación se exponen ambos:

I. Objetivo del problema:

El objetivo del problema es determinar la ruta óptima para un vehículo de distribución. Una ruta se define como la secuencia de llegada a cada uno de los clientes saliendo a partir de un centro de distribución y regresando al mismo al final de la ruta. Se define una ruta óptima como aquella que logre visitar todos los clientes de la manera más eficiente posible en términos del costo (la distancia o el tiempo) y a la vez lo haga considerando el aspecto de las ventanas de horario definidas para dar servicio y atención a cada cliente.

II. Condiciones de operación para el problema:

1. Se requiere obtener la ruta óptima que debe cubrir un vehículo de carga con capacidad finita y estando localizado en un centro de distribución.
2. El vehículo, saliendo desde el centro de distribución, debe atender a un conjunto de clientes geográficamente dispersos y luego regresar al punto de origen (centro de distribución).

3. Entre cada uno de los clientes se tiene definida una matriz de costo que resulta de establecer el nivel de gasto o distancia requerida para trasladarse de un cliente a otro.
4. Entre cada uno de los clientes se tiene definida una matriz de tiempo discreto que establece el número de secuencias de arribo que se requieren para trasladarse de un cliente a otro.
5. Para cada uno de los clientes existe un requerimiento definido por los siguientes elementos:
 - a. Un volumen de producto a entregar.
 - b. Un volumen de producto a recoger.
6. La cantidad de tiempo requerido para realizar ambas operaciones (entrega y recolección) en cada cliente se considera razonablemente el mismo por tratarse de cargas de trabajo similares.
7. Las ventanas de horario identificadas para cada cliente están definidas por una hora de apertura y por una hora de cierre la cual puede tener diferente amplitud dependiendo de las características de cada cliente.
8. La amplitud de la ventana de horario en cada cliente es igual a la diferencia entre la hora de cierre y la hora de apertura para la atención del cliente.
9. Las ventanas de horario identificadas en cada cliente pueden ser negociables (o re-definidas) de acuerdo a la conveniencia económica en el recorrido de la ruta que pueda presentarse al haber otros clientes geográficamente cercanos y cuyas ventanas de horario sean diferentes.
10. Las visitas a ser hechas a cada uno de los clientes, deberán ser aplicadas rigurosamente dentro de la ventana de horario que haya sido definida en el punto anterior.
11. No está permitido llegar antes de la hora de apertura del cliente. Es decir, no se pueden considerar tiempos de espera en el punto de venta hasta el tiempo de la apertura del cliente de acuerdo a la ventana de horario.
12. Tampoco es factible llegar después de la hora de cierre del cliente. En tal caso la ruta se considera inválida y es necesario evaluar otras opciones.
13. Por lo mencionado en los puntos 11 y 12, en sustitución se recurre a re-negociar las ventanas de horario que hagan falta y así aprovechar al máximo el tiempo operativo de la ruta (ver punto 9).
14. De acuerdo a la secuencia de ruteo establecida para el vehículo, se tendrá una descarga de producto a entregar en el cliente y una carga a recoger del cliente. En todo momento deberá de ser respetada la capacidad de carga del vehículo.

Hasta aquí el planteamiento formal del problema de investigación. A continuación, en el siguiente apartado procederemos a la modelación matemática del mismo. Más adelante se expondrán las referencias bibliográficas de trabajos de investigación que han abordado previamente el tema.

6.3 Modelación matemática del problema de investigación.

El desarrollo del modelo matemático para un problema es un buen punto de inicio. De hecho a partir de una formulación hecha en términos de programación lineal, es posible llegar a estar cerca de la solución de muchos tipos de problemas. No obstante, debemos advertir que no resulta así en el caso de nuestro problema de investigación por tratarse de un problema de optimización discreta NP-Hard. Aunque para un problema de programación lineal existe una cantidad infinita de soluciones, no obstante existen ya los métodos para explotar la estructura de este tipo de problemas y lograr convertir este espacio de soluciones infinito en uno finito. En el corazón de estos métodos se tiene al método “Simplex”.

Por otro lado, para los problemas de optimización discreta se tienen espacios de solución que aunque son inmensos resultan ser finitos. Paradójicamente, para el caso de esta clase de problemas no se dispone del mismo avance algorítmico tal como el que se verifica en el caso de los problemas de programación lineal. La contraparte de la programación lineal es la programación mixta entera (MIP). Esta fue introducida en la década de los 60's¹⁰². Se tuvieron intentos de aplicación en aquel entonces, sin embargo en ese momento el software y el hardware computacional para la solución de este tipo de problemas era apenas incipiente. Fue probablemente, la falta de capacidad y de disponibilidad de software y hardware lo que durante décadas propició la falta de aplicación.

No obstante lo anterior, podemos ahora mencionar que la situación en cuanto a estas capacidades de software y de hardware computacional han cambiado dramáticamente sobre todo a partir de 1999, año en cual se desarrollaron nuevas estrategias y mejoras tanto en los algoritmos de programación lineal así como también en los algoritmos de programación mixta entera¹⁰³.

En nuestro problema del SPDP-sTW, se requiere la construcción completa de una di-gráfica (o gráfica bi-dimensional) determinada por $G=(V,A)$, donde:

1. V esta constituido por el conjunto de los nodos de la red de distribución $\{1, \dots, n\}$.
2. A esta constituido por el conjunto de los arcos que forman la red de distribución $\{(i,j): i,j \in V\}$.

¹⁰² Pawda, Juan. (2000), Modelos de Investigación de Operaciones, 1era.Ed, Limusa, México, pag 73.

¹⁰³ Bixby, Robert. (1999), MIP: Theory and practice closing the gap, Ilog Cplex Division, Department of Computational and Applied Mathematics, Rice University, Houston EUA, pag 3.

3. Se tiene definida una matriz $C_{i,j} = 0$ que determina el costo para trasladarse de cada nodo "i" a cada nodo "j" $\{(i,j) \in A\}$ y definiendo que $C_{i,i} = ?$ para cada $i \in V$.
4. Además se tiene definida una matriz $T_{i,j} = 0$ que determina el tiempo para trasladarse de cada nodo "i" a cada nodo "j" $\{(i,j) \in A\}$ y definiendo que $T_{i,i} = ?$ para cada $i \in V$.
5. El objetivo es formar un ciclo Hamiltoniano (tour o recorrido cerrado) de extensión o costo mínimo que logre recorrer a cada uno de los nodos visitándolos una vez a cada uno de ellos. Matemáticamente tenemos entonces que se requiere obtener una $G=(V,A)$ de tal manera que $|A| = n$ asegurando que cada uno de los nodos $v_1, v_2, \dots, v_n \in V$ y buscando como objetivo minimizar $\sum_{(i,j) \in A} C_{ij}$.

A continuación se muestra la modelación matemática detallada de nuestro problema de investigación.

Variables de Entrada (1 de 2):

D_i = Demanda de producto entrante hacia el nodo "i".

$$\begin{cases} = 0 & \text{El nodo "i" solo sirve como mecanismo de transferencia topológica.} \\ \neq 0 & \text{El nodo "i" SI es un cliente para logística de entrega.} \end{cases}$$

O_i = Devolución o retorno de producto saliente desde el nodo "i".

$$\begin{cases} = 0 & \text{El nodo "i" solo sirve como mecanismo de transferencia topológica.} \\ \neq 0 & \text{El nodo "i" SI es un cliente para logística inversa.} \end{cases}$$

Restricciones topológicas de la red (Simétrica o Asimétrica):

C_{ij} = Costo / Tiempo requerido para ir desde el nodo "i" hacia el nodo "j"

E = Capacidad de carga de la unidad de transporte.

Variables de Entrada (2 de 2):

Restricciones de Ventana de Tiempo (Time Window):

A_i = Tiempo de llegada más temprano permitido para el nodo "i".

Donde: $A_i = 0$, para los nodos de transferencia

o para nodos sin restricciones de TW.

P_i = Tiempo de atención / procesamiento para el nodo "i".

Donde: $P_i = 0$, para los nodos de transferencia

B_i = Tiempo de llegada más tarde permitido para el nodo "i".

Donde: $B_i = +Inf$, para los nodos de transferencia

o para nodos sin restricciones de TW.

Variables de Salida (1 de 2):

$$X_{ijk} \geq 0, \leq 1, \text{ent} \quad \begin{cases} 0 \rightarrow \text{NO se va desde el nodo "i" hacia} \\ \text{el nodo "j" en la secuencia "k"} \\ 1 \rightarrow \text{SI} \end{cases}$$

$$T_j \geq 0 \quad \{\text{Tiempo de llegada al nodo "j"}\}$$

$$W_j \geq 0 \quad \{\text{Tiempo de procesamiento en el nodo "j"}\}$$

$$S_j \geq 0 \quad \{\text{Tiempo de salida del nodo "j"}\}$$

$$Z_j \geq 0 \quad \{\text{Tiempo de traslado hacia el nodo "j"}\}$$

Variables de Salida (2 de 2):

$$AA_j \geq 0 \quad \{\text{Tiempo de llegada al nodo "j" antes de la hora de apertura.}\}$$

$$DA_j \geq 0 \quad \{\text{Tiempo de llegada al nodo "j" después de la hora de apertura.}\}$$

$$AC_j \geq 0 \quad \{\text{Tiempo de llegada al nodo "j" antes de la hora de cierre.}\}$$

$$DC_j \geq 0 \quad \{\text{Tiempo de llegada al nodo "j" después de la hora de cierre.}\}$$

$$Q_j \geq 0, \leq 1, \text{ent} \quad \begin{cases} 0 \rightarrow \text{NO se cumple el "Time Window" del cliente "j"} \\ 1 \rightarrow \text{SI} \end{cases}$$

$$IN_j \geq 0 \quad \{\text{Carga en "kgs" que se va a entregar en el cliente "j"}\}$$

$$OUT_j \geq 0 \quad \{\text{Carga en "kgs" que se va a recojer en el cliente "j"}\}$$

$$F_j \geq 0 \quad \{\text{Carga en "kgs" del camión a la salida del cliente "j"}\}$$

Restricciones (1 de 4):

$$\sum_{i=1}^{n1} \sum_{j=1}^{n2} X_{ijk} = 1 \quad \left\{ \begin{array}{l} \text{para cada secuencia solo puede} \\ \text{existir un solo movimiento desde un} \\ \text{nodo "i" hacia un nodo "j"} \end{array} \right\}$$

para $k = 1..n3$

$$\sum_{i=1}^{n1} \sum_{k=1}^{n3} X_{ijk} \geq Y_j \quad \left\{ \begin{array}{l} \text{para cada nodo de entrada "j" al cual se tenga} \\ \text{que llegar, deberá ser proveniente de cualquier} \\ \text{nodo "i" y en cualquier secuencia "k"} \end{array} \right\}$$

para $j = 1..n2$

$$\sum_{i=1}^{n1} X_{ijk} = \sum_{i=1}^{n1} X_{jik+1} \quad \{\text{Eliminación de Sub-Tours}\}$$

para $j = 1..n2,$
 $k = 1..n3$

Restricciones (2 de 4):

$$T_j = S_{j-1} + Z_j \quad \left\{ \begin{array}{l} \text{para cada nodo "j", su tiempo de llegada} \\ \text{es igual a l tiempo d e salida d el nodo an terior} \\ \text{más el tie mpo de tra slado haci a el nodo "j".} \end{array} \right\}$$

para $j = 1..n2$

$$S_j = T_j + W_j \quad \left\{ \begin{array}{l} \text{para cada nodo "j" , su tiempo de salida} \\ \text{es igual a su tiempo de llegada más el} \\ \text{tiempo de proceso en el nodo "j".} \end{array} \right\}$$

para $j = 1..n2$

$$W_j = \sum_{i=1}^{n1} \sum_{k=1}^{n3} X_{ijk} * P_j \quad \left\{ \begin{array}{l} \text{Tiempo de proceso en el nodo "j",} \\ \text{sin importar desde cual nodo "i" provenga} \\ \text{y en que secuencia "k" lo haga.} \end{array} \right\}$$

para $j = 1..n2$

$$Z_j = \sum_{i=1}^{n1} \sum_{k=1}^{n3} X_{ijk} * C_{ij} \quad \left\{ \begin{array}{l} \text{Tiempo de traslado hacia el nodo "j",} \\ \text{desde el nodo "i" y en la secuencia "k"} \end{array} \right\}$$

para $j = 1..n2$

Restricciones (3 de 4):

$$A_j - T_j = AA_j - DA_j \quad \left\{ \begin{array}{l} \text{tiempo de llegada al nodo "j" antes o después} \\ \text{de la hora de apertura del "Time Window"} \end{array} \right\}$$

para $j = 1..n2$

$$B_j - T_j = AC_j - DC_j \quad \left\{ \begin{array}{l} \text{tiempo de llegada al nodo "j" antes o después} \\ \text{de la hora de cierre del "Time Window"} \end{array} \right\}$$

para $j = 1..n2$

$$\frac{AA_j + DC_j}{LI \quad LS} \quad 1 - Q_j \leq AA_j + DC_j \quad \text{para } j = 1..n2$$

$$1 \quad \infty \quad Q_j = 0 \quad \infty(1 - Q_j) \geq AA_j + DC_j$$

$$0 \quad 0 \quad Q_j = 1 \quad \text{Cumplimiento del Time Window para el Nodo "j"}$$

Restricciones (4 de 4):

$$IN_j = \sum_{i=1}^{n1} \sum_{k=1}^{n3} X_{ijk} * D_j \quad \left\{ \begin{array}{l} \text{Cantidad de carga a} \\ \text{entregar en el nodo "j"}. \end{array} \right\}$$

$$OUT_j = \sum_{i=1}^{n1} \sum_{k=1}^{n3} X_{ijk} * O_j \quad \left\{ \begin{array}{l} \text{Cantidad de carga a} \\ \text{recojer en el nodo "j"}. \end{array} \right\}$$

$$F_j = F_{j-1} - IN_j + OUT_j \quad \left\{ \begin{array}{l} \text{La carga en "kgs" con la cual se sale del} \\ \text{cliente "j" es igual a la carga con la cual} \\ \text{se llega, menos la que se va a entregar} \\ \text{más la que se va a recojer.} \end{array} \right\}$$

$$\therefore F_j \leq E \quad \left\{ \begin{array}{l} \text{La carga en "kgs" a la salida del cliente "j",} \\ \text{no debe exceder a la capacidad de carga del camión.} \end{array} \right\}$$

Funciones Objetivo (1 de 2)

Enfoque al cumplimiento de las Ventanas de Horario:

$$FO_{max} : \sum_{j=1}^{n2} Q_j \quad \left\{ \begin{array}{l} \text{Maximizar la cantidad de clientes con} \\ \text{cumplimiento en el "Time Window"} \end{array} \right\}$$

$$FO_{min} : \sum_{j=1}^{n2} AA_j + DC_j \quad \left\{ \begin{array}{l} \text{Minimizar el tiempo total de} \\ \text{incumplimiento del "Time Window"} \\ \text{de todos los clientes en el sistema.} \end{array} \right\}$$

$$FO_{min} : MAY \quad \left\{ \begin{array}{l} \text{Minimizar el máximo incumplimiento de todos} \\ \text{los "Time Window" del Sistema.} \end{array} \right\}$$

$$FO_{min} : S_n \quad \left\{ \text{Minimizar el tiempo de recorrido de toda la ruta} \right\}$$

Funciones Objetivo (2 de 2)

Enfoque al factor económico de la ruta

$$FO_{min} = \sum_{i=1}^{n1} \sum_{j=1}^{n2} \sum_{k=1}^{n3} X_{ijk} * C_{ij} \quad \left\{ \begin{array}{l} \text{Minimizar la cantidad de distancia, tiempo} \\ \text{o costo económico implicado para recorrer} \\ \text{la totalidad de los clientes de la ruta.} \end{array} \right\}$$

El Proyecto de Investigación se centra en este tipo de consideración.

6.4 Aproximaciones para la solución del SPDP-sTW vía algoritmos basados en el TSP.

A partir del apartado 6.4 al 6.7, el postulante expone la revisión bibliográfica en lo referente a los trabajos de investigación previos que han sido desarrollados para resolver el problema de investigación. En principio, es conveniente puntualizar que dicha exposición no se presentó como parte del marco conceptual debido a la especificidad del problema de investigación que recién hemos formalmente planteado en el apartado 6.2. Por tal motivo, estos apartados (6.4 al 6.7), pueden ser vistos como parte complementaria del marco conceptual pero ya ahora enfocado al problema específico de investigación.

Al revisar cada uno de los apartados (6.4 al 6.7) se podrá comprobar que aún en este momento, el postulante hará su exposición bibliográfica partiendo de variantes similares al problema específico referido al SPDP-sTW (TSP en el apartado actual), hasta llegar a la variante específica en el apartado 6.6 y finalmente para llegar a la exposición de lo que el postulante considera, el estado del arte en cuanto a la técnica de solución empleada, la cual se expone en el apartado 6.7.

Para el problema de investigación SPDP-sTW, una buena referencia para establecer una estrategia de ataque es a partir de las investigaciones que se hayan desarrollado con problemas de investigación parecidos. Garey y Johnson demostraron en 1989, que aún la tarea de encontrar una solución factible (ciclo Hamiltoniano) para el problema del TSP resulta ser un problema NP-Hard, no obstante esto resulta ser en términos relativos más fácil que lograr encontrar su solución óptima lo cual resulta por consecuencia aún más difícil ¹⁰⁴. En términos generales podemos decir que el problema TSP ha sido resuelto mediante métodos de solución exacta con razonable éxito de acuerdo al tamaño de las instancias que fueron experimentadas.

El método de solución exacta mayormente empleado ha sido históricamente el método de Ramificación y Corte (Branch & Cut en inglés, de aquí en adelante se le referenciará con las siglas BC). Así entonces, resulta sustentable establecer al algoritmo BC, como el método que será utilizado para calcular la cota inferior que será utilizada para contrastar la solución que será ofrecida por la propuesta de investigación doctoral basada en el algoritmo genético. Dicha definición del BC como cota inferior será expuesta más adelante en el apartado 7.3.

¹⁰⁴ Garey, M; Johnson, D. (1989), *Computers and Intractability: a Guide to the Theory of NP-Completeness*, Freeman Ed, San Francisco EUA, pag 67-89.

Una de las propuestas de investigación basadas en el BC para la solución del TSP considera la posibilidad de transformar el mismo problema del TSP en un problema de asignación (léase AP). Lo anterior se logra a través del relajamiento de las restricciones de conectividad (se omiten las restricciones para la eliminación de sub-tours) en el problema básico del TSP y convirtiéndolo en un problema de AP. Lo anterior equivale a no incluir en el sistema de restricciones $Ax \leq b$ todos aquellos cortes (o hiperplanos) que aseguren la eliminación de subtours en el circuito Hamiltoniano. Así, el problema del AP simplemente trata de encontrar el conjunto de sub-tours (tours no conectados) de tal forma que se asegure incluir a todos los nodos incluidos en V al mínimo costo.

Lo interesante del enfoque basado en la solución del problema AP aplicado al TSP (y también al PDP-TW), radica en que en el primero existe una propiedad poco común entre los problemas de redes y es que en principio las restricciones de integralidad no requieren ser impuestas ya que su solución mediante programación lineal siempre arrojará implícitamente soluciones enteras. Lo anterior aplicado al problema básico del TSP, equivaldría a disponer de un método de solución basado tan sólo en programación lineal dejando de requerir la aplicación de un algoritmo de enumeración parcial basado en el BC. Así, si la solución obtenida en el problema del AP mediante la relajación del problema original incluye un solo ciclo Hamiltoniano (cerrado), pues entonces se tiene ya la solución óptima. Desafortunadamente esto rara vez ocurre en la práctica.

La dificultad del enfoque anterior se presenta al intentar integrar los cortes en el sistema $Ax \leq b$ que vayan restringiendo la aparición de sub-tours en la solución. Normalmente lo anterior no sería un problema insalvable ya que son bien conocidos los algoritmos requeridos para generar estas restricciones (o hiperplanos). El problema se presenta debido a que al agregar dichas restricciones se pierde para siempre la propiedad inherente del AP que permitía solucionar sus variables enteras mediante tan solo la aplicación de programación lineal.

En cualquier caso y habiendo asumido la pérdida de la propiedad anterior, es importante considerar que la solución ofrecida por el planteamiento del AP mediante programación lineal, resulta ser la cota inferior para el problema original del TSP. Esto quiere decir, que en la misma corrida anterior y sin requerir de esfuerzo computacional adicional, es posible disponer de la matriz de costos reducida (o también llamada solución dual) con la cual es posible verificar la conveniencia relativa (penalización adicional) que cada uno de los arcos (i,j) tienen al ser incluidos en la solución del problema.

Lo anterior se puede explicar de una manera aún más formal diciendo que la cota inferior LB para un arco $x_{ij} = 1$ se recalcula como: $LB \text{ (para } x_{ij}=1) = LB + cij$, donde cij es el costo reducido o penalización implicada por agregar el arco x_{ij} en la solución final correspondiente al ciclo hamiltoniano. Por lo tanto resulta obvio decir, que aquellas variables x_{ij} que tengan calculado un costo reducido cij menor, serán las mismas que tengan mayor probabilidad de aparecer en la solución final. Desafortunadamente respecto a la exposición desarrollada hasta aquí, es necesario decir que según el trabajo de Dell'Amico y de Martello en 1997, se concluye que en términos generales las cotas obtenidas del problema original mediante su relajación vía la aplicación del AP no resultan ser eficientes en el sentido de estar ajustadas al problema original del TSP, matemáticamente hablando ¹⁰⁵. Por tanto, es predecible que ocurra lo mismo para nuestro problema de investigación SPDP-sTW.

Como complemento a la estrategia anterior desarrollada por Dell'Amico y Martello, en 1996 Balas y Simonetti, implementaron una metodología basada en el trabajo previo que Lawler y Lenstra desarrollaron en 1986 para la generación de cortes que lograban restringir la aparición de sub-tours en el TSP (sub-tours elimination constraints) ¹⁰⁶. Esta última propuesta, no establecía su estrategia a partir de una relajación basada en el problema del AP sino tan solo relajaba las restricciones para la eliminación de los sub-tours.

Una última investigación fue desarrollada en 1999 cuando Focacci y Milano implementaron una relajación del TSP mediante un AP pero incluyeron una estrategia diferente para implementar los cortes requeridos para restringir los sub-tours. En lugar de incluir los cortes como parte del sistema de restricciones $Ax \leq b$, decidieron hacerlo como parte de la función objetivo a través de la aplicación de técnicas de relajación de Lagrange ¹⁰⁷.

¹⁰⁵ Dell'Amico, M; Martello, S. (1997), Linear assignment: Annotated Bibliographies in Combinatorial Optimization, Wiley & Sons editors, EUA, pag 355–371.

¹⁰⁶ Balas, E; Simonetti N. (1996), Linear time dynamic programming algorithms for some new classes of restricted TSP's, "Management Science Research Report 617", Graduate School of Industrial Administration, Carnegie Mellon University, EUA.

¹⁰⁷ Focacci, F; Milano, M. (1999), Solving Tsp with time windows with constraints, "INFORMS Journal of Computing, ICLP 99", International Conference on Logic Programming, EUA.

6.5 Aproximaciones para la solución del SPDP-sTW basadas en el TSP-TW y en el VRP-TW.

No hay duda de que el problema TSP-TW es mucho más parecido a nuestro problema de investigación que el TSP. Brevemente podemos resumir que varios algoritmos han sido propuestos para el TSP-TW. Ellos van desde la aplicación de heurísticas tales como las de inserción de Lin-Kernighan. Otros algoritmos están basados en la aplicación de métodos de solución exacta basados en la aplicación de BC y más aun otras en la aplicación de la programación dinámica con relajación de estado de transición. De lo que si no hay duda es que el TSP-TW es un problema NP-Hard, lo cual implica que no exista un algoritmo que pueda resolver el problema en un tiempo de ejecución polinomial y que al mismo tiempo la programación dinámica tradicional esté propensa a fallar ¹⁰⁸.

Ahora bien, resulta que problema más parecido al SPDP-sTW es el VRP-TW. Ambos son considerados problemas relevantes en la investigación actual y con una herencia compartida respecto al problema del TSP. Así pues, identificar que se ha desarrollado en el estudio del VRP-TW es una buena estrategia. En 1984 Solomon desarrolló 87 instancias de prueba para el VRP-TW. De todas ellas, la más grande es de 100 clientes ¹⁰⁹. De hecho, de 17 instancias que aún permanecían sin poder ser resueltas, 10 de ellas fueron apenas resueltas en 1999 por Cook y Rich de la Universidad de Rice en Houston ¹¹⁰. No obstante todo el esfuerzo que se ha desplegado en términos de investigación en el VRP-TW en estos últimos años, aún existen 7 instancias que no han podido ser resueltas en la literatura científica.

Es importante precisar que la estrategia de solución implementada por Cook y Rich se basa en un algoritmo cuya base es programación lineal y que se apoya en un motor de búsqueda el cual a su vez está apoyado en un generador de cortes (o hiperplanos) derivados de un algoritmo de BC. El algoritmo de BC implementado tiene la particularidad de que incluye un procedimiento para la generación de columnas aplicadas de manera incremental y progresiva sobre el problema maestro. A este enfoque de generación incremental de columnas, se le denomina en inglés Branch & Price por tratarse de una derivación del método BC tradicional. Esto último se explica más adelante de manera breve.

¹⁰⁸ Fisher, Marshall. (1995), Overview over optimization models in transportation, Handbooks in Operations Research and Management Science, North Holland, EUA, Pag 10.

¹⁰⁹ Solomon, M. (1984), On the worst-case performance of some heuristics for the vehicle routing and scheduling problem with time window constraints, "Report 83-05-03", The Wharton School, University of Pennsylvania EUA.

¹¹⁰ Cook, W; Rich, Jennifer. (1999), A parallel cutting-plane algorithm for the vehicle routing problem with time windows, Computational and Applied Mathematics Rice University, Houston EUA, pag 5.

La estrategia utilizada por Cook y Rich para la identificación de las restricciones (o cortes) se deriva de las heurísticas de eliminación de sub-tours de diversos tipos tales como :

1. $k=1$: desarrollados por Dantzig, Fulkerson, y Johnson en 1954.
2. $k=2$: desarrollados por Kohl, Desrosiers, Madsen, Solomon en 1992.
3. “ k -path” para $k \geq 3$: desarrollado por Karger en 1993 ¹¹¹.

El enfoque consiste en iniciar con un problema maestro limitado en cuanto a la cantidad de rutas que van a ser evaluadas en la primera iteración (recordemos que el VRPTW trata de un problema con múltiples rutas). Así de esta manera, se aplica un procedimiento recursivo en el cual después de ejecutar el problema maestro mediante programación lineal, se utilizan entonces los costos reducidos de los arcos (i,j) (valores duales) a fin de ir progresivamente incorporando el resto de los nodos y por ende de las rutas en el problema maestro. Cada nueva ruta que se va incorporando, representa una nueva columna a ser adicionada en la base numérica del problema. A este método en el que se combina la utilización de generación de cortes a la vez que los costos reducidos de las variables, se le denomina “Método de ramificación y costo” (o en inglés Branch & Price, BP). El método BP ha resultado ser una buena alternativa para resolver los problemas referentes al VRP-TW. Mayor detalle respecto al método puede ser revisado en la obra de Cook ¹¹².

Es oportuno finalizar el apartado mencionando que básicamente en el método BP, cada una de las columnas que se van agregando a la base numérica del problema resulta ser una ruta asociada a cada uno de los vehículos que están siendo integrados de manera gradual al problema principal. Dado que nuestro problema de investigación SPDP-sTW está enfocado en dar tratamiento a un solo vehículo, por tanto es predecible que el BP no resulte ser una estrategia de solución conveniente a ser propuesta.

¹¹¹ Karger, D. (1993), Global min-cuts in RNC and other ramifications of a simple min-cut algorithm, “Proceedings of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms”, EUA, pag 84-93.

¹¹² Cook, W. (1999), A parallel cutting-plane algorithm for the vehicle routing problem with time windows, Computational and Applied Mathematics Rice University, Houston Texas, pag 2.

6.6 Revisión algorítmica referente a trabajos de investigación previos para el problema PDP-TW.

Las investigaciones en el problema PDP-TW en términos generales iniciaron hace apenas 10 años. Aunque más adelante se establecerán las diferencias específicas que delimitan nuestro problema de investigación (SPDP-sTW) respecto al problema general del PDP-TW, a continuación se muestra una breve cronología de las investigaciones más relevantes realizadas con respecto al tratamiento de este problema. Esto tiene el propósito de establecer los diversos enfoques que han sido explorados para dar solución al problema en cuestión:

1. El problema del PDP-TW fue apenas por primera vez atendido en el año de 1993 por el trabajo de Dumas, Desrosiers y Soumis. Ellos propusieron varias estrategias para atacar el problema mediante una formulación matemática basada en los problemas típicos de flujo de redes .
2. Más adelante tenemos el trabajo de Applegate y Bixby, los cuales en 1994 se dieron a la tarea de investigar y explotar la estructura matemática del problema con la finalidad de lograr obtener el sistema de restricciones o cortes que mejor lograba resolver este tipo de problemas.
3. Bruggen, Lenstra y Schuur en 1993 experimentan con dos alternativas para la solución del problema. El primer experimento fue a través de la aplicación de un procedimiento de búsqueda local basado en el método de intercambio de Lin-Kernigham. El segundo se desarrollo a partir de la aplicación de la meta-heurística “Enfriamiento Simulado” (Simulated Annealing) desarrollado en 1985 por Kirkpatrick y Lawler. Ambos experimentos fueron probados exitosamente con instancias de no más allá de 50 nodos y con ventanas de horario estrechas. “Arriba de 50 nodos y con ventanas de horario amplias, los resultados no fueron tan buenos y la principal desventaja empezó a ser el tiempo computacional ya que el procedimiento resultó ser sumamente lento requiriendo arriba de 10 minutos de tiempo computacional” ¹¹³.

¹¹³ Mitrovic, Snezana. (1998), Pickup and Delivery Problem with Time Windows “Technical Report SFU CMPT TR 1998-12”, Canada, pag 31.

4. Van Eijl en 1995 propuso un algoritmo de cortes basado en el BC. Sus resultados computacionales fueron probados en instancias hasta de 15 nodos y con tiempos de ejecución computacional muy elevados en términos relativos, aunque no se reporta en la investigación algún valor específico de tiempo ¹¹⁴.
5. Coth y Vigo en 1995-1996 presentan otro algoritmo también basado en la aplicación de un procedimiento meta-heurístico. En esta ocasión se aplica el procedimiento desarrollado por Glover en 1987 denominado “Búsqueda Tabu” (o Tabú Search). Los resultados son modestos aunque el principal problema seguía siendo que el procedimiento requería de considerables tiempos de ejecución computacional. El algoritmo fue implementado en dos fases, la primera fase correspondía a la aplicación de una heurística de inserción. La segunda fase corresponde a la fase de mejoramiento mediante la aplicación de la meta-heurística “Tabu Search”. Los tiempos de ejecución reportados para instancias clasificadas como de “gran escala” (arriba de 100 nodos) fueron de alrededor de una hora ¹¹⁵.
6. Resulta obligado referenciar las aportaciones de Dumas, Desrosiers y Solomon, los cuales en 1995 presentan un algoritmo basado en la aplicación del método de descomposición de Dantzig-Wolfe (1956). Este método de descomposición (o también llamado de “generación de columnas”), tiene su utilidad en la posibilidad de integrar las teorías de programación lineal de Dantzig y Wolfe en la obtención de cotas más ajustadas para la solución de problemas de programación entera. Lo anterior redundaba en la obtención de un árbol de búsqueda mucho más pequeño a tener que ser explorado por los algoritmos de BC. En términos generales, se le ha referenciado a esta propuesta algorítmica como la mejor alternativa disponible para aproximar soluciones al problema. Los experimentos computacionales reportan instancias de entre 100 y 200 nodos requiriendo tiempos computacionales de entre 20 y 30 minutos ¹¹⁶.

¹¹⁴ Eijl Van, C. (1995), A polyhedral approach to the delivery man problem, “ Technical Report 95-19”, Department of Mathematics and Computer Science, Eindhoven University of Technology, The Neatherlands, pag 12-14.

¹¹⁵ Mitrovic, Snezana. (1998), Pickup and Delivery Problem with Time Windows, “ Technical Report SFU CMPT TR 1998-12”, Canada, pag 32-33.

¹¹⁶ Dumas, Y; Desrosiers, J; Gelinas, E; Solomon, M. (1995), An algorithm for the traveling salesman problem with time windows, “Operations Research 43(2)”, EUA, pag 23-25.

7. En 1996 Balas y Simonetti, presentan un algoritmo de solución exacta basado en programación dinámica. Su estrategia se basa primordialmente en la implementación de algoritmos para reducción del espacio solución vía relajamiento de los estados de transición típicos en la aplicación de las técnicas de programación dinámica. La única consideración es que el algoritmo solo funciona adecuadamente en instancias de hasta 40 o 50 nodos y con poca incidencia en el traslape de las ventanas de horario. Los tiempos reportados en el experimento son arriba de 2 horas de ejecución computacional¹¹⁷. Dichas investigaciones apuntan a concluir que para poder aplicar el algoritmo de solución exacta basado en programación dinámica se requiere que exista poca probabilidad de que se presenten conflictos entre las ventanas de horarios de los clientes a una misma hora. Se concluye en el reporte que la principal desventaja de la programación dinámica es la inherente influencia que existe en la forma en como se implemente la discretización del tiempo en el algoritmo lo cual limita su aplicabilidad debido a las restricciones de recursos computacionales.
8. En 1997 Mingozzi, Bianco, y Ricciardelli propusieron una metodología híbrida basada en la aplicación de BC para la definición de las acotaciones del espacio solución pero complementándolo con una variante de la programación dinámica en la cual se aplica una relajación de los estados de transición para explotar el espacio solución obtenido previamente en la fase de BC. Aquí se presentan instancias resueltas hasta de 120 nodos aunque no se incluyen los resultados de los tiempos computacionales¹¹⁸.
9. Ascheuer, Jünger y Reinelt en el año 2000 aplican una estrategia diferente. En esta ocasión ellos utilizan un algoritmo generador de cortes (hiperplanos) basado en el BC. El algoritmo fue aplicado en una variante del TSP pero solo con restricciones de precedencia. Es bien sabido que las restricciones de precedencia y las restricciones de ventana de horario son bastante similares. La estrategia basada en el BC fue exitosa (tiempos debajo de 20 minutos). No obstante, en este experimento no fue posible identificar cual sería el resultado al aplicar el algoritmo propuesto a un problema modificando las restricciones de

¹¹⁷ Balas, E; Simonetti, N. Linear time dynamic programming algorithms for some classes of restricted tsp's Technical Report MSRR No. 617; Carnegie Mellon University; Pittsburgh EUA 1997; pag 10-11.

¹¹⁸ Mingozzi, A; Bianco, L; Ricciardelli, S. (1997), Dynamic programming strategies for the travelling salesman problem with time windows and precedence constraints, "Operations Research No. 45", EUA, pag 277.

precedencia por las del ventana de horario ¹¹⁹. Dichos resultados se comprobarían un año después cuando Ascheuer, Fischetti y Grottschel desarrollarían un experimento para ese propósito específico (ver siguiente inciso 10).

10. El último trabajo revisado en la investigación bibliográfica fue desarrollado en el año 2001 por Ascheuer (Alemania), Fischetti (Italia), y Grottschel (Alemania). Esta investigación se centra en la aplicación de una metodología basada en el BC pero aplicándolo para una variante asimétrica del TSP (es decir en el ATSP). Ellos inician el planteamiento de su investigación sobre la base de una amplia revisión bibliográfica de los trabajos previos hechos sobre el problema del ATSP-TW. Como es de esperarse, el trabajo de ellos se concentra en el tratamiento de las ventanas de horario (TW) en el ATSP. Los resultados de su investigación se presentarán en el siguiente apartado 6.7.

¹¹⁹ Ascheuer, N; Jünger, M; Reinelt, G. (2000), A branch & cut algorithm for the asymmetric Traveling Salesman Problem with precedence constraints, "Computational Optimization and Applications 17(1)", EUA, pag 61–84.

6.7 Identificación del estado del arte para el problema de investigación.

Como ya se mencionó antes, el problema del TSP-TW es un problema relativamente cercano en definición al PDP-TW. Nuestro punto de partida será el último trabajo presentado en la cronología expuesta del apartado anterior. Ascheuer, Fischetti, y Grotschel en el 2001 mencionan en su trabajo que resulta interesante la poca atención que se le ha brindado a este problema en particular. Especialmente mencionan, que los algoritmos de solución exacta basados en el BC han tenido un protagonismo menor tanto para la versión simétrica (TSP-TW) así como para la versión asimétrica (ATSP-TW).

Como antecedente en este rubro mencionan a Baker, el cual en la década de los 80's resolvió mediante algoritmos de solución exacta de BC instancias de hasta 50 nodos con ventanas de horario moderadamente cerradas (matemáticamente hablando, entre más cerradas sean las ventanas de horario más pequeño es el espacio solución a ser explorado y por tanto más fácil es el problema en resolver). Además estas instancias estaban caracterizadas por ventanas de horario con un porcentaje bajo de traslape¹²⁰. No se dispone de registros acerca de la velocidad computacional que se obtuvo en el experimento de Baker tomando en cuenta el avance de los procesadores matemáticos de aquella época.

De la revisión anterior, ellos parten de la existencia hasta la fecha de tres modelos basados en programación mixta entera (MIP), y solucionados mediante la aplicación de algoritmos de BC. Hasta antes de su investigación, ellos afirman no tener conocimiento de cual de estos tres modelos es el que ofrece mejor perspectiva computacional. Así pues, su proyecto de investigación consistió en determinar un análisis comparativo entre estos tres para comprobar cual era el mejor modelo. Ascheuer, Fischetti, y Grotschel precisan que para el TSP-TW, no resulta fácil comparar diferentes algoritmos para su solución ya que en principio no se dispone de instancias que puedan servir como “benchmark” estándar del problema en cuestión.

Habitualmente la dificultad computacional para la solución de las instancias de los problemas ha sido tipificada en términos del tamaño (cantidad de nodos) del problema. Sin embargo es obvio precisar que la dificultad de este tipo de problemas depende fuertemente de la estructura de las ventanas de horario que se definan. En general, el resultado de los experimentos con el TSP-TW hechos por Ascheuer, Fischetti, y Grotschel, arrojaron que este problema es particularmente difícil de resolver para aquellas instancias en las que la cantidad de nodos activos que contengan restricciones de ventana de

¹²⁰ Ascheuer, N; Fischetti, M; Grotschel, M. (2001), Solving ATSP with time windows by branch-and-cut , Springer-Verlag , EUA.

horario estén por arriba del 50%. En lo referente al tamaño de los problemas que fueron probados en el experimento solo uno de los tres modelos revisados pudo resolver instancias de más de 69 nodos ¹²¹.

Una vez verificado cual de los tres modelos es el mejor, entonces Ascheuer, Fischetti, y Grotscel desarrollaron una versión especializada sobre la base del mejor modelo de los tres. En esta versión, ellos implementaron una serie de al menos 15 procedimientos heurísticos con la finalidad de mejorar aún más el desempeño del algoritmo original. Se corrieron instancias de hasta 233 nodos. La instancia de 69 nodos requirió de 5.95 minutos de tiempo computacional. El resto de todas las instancias mayores ocuparon más de 5 horas en la solución computacional (no se especifica cuanto más de 5 horas); excepto una de 127 nodos que pudo resolverse en menos de 5 horas.

Ascheuer, Fischetti, y Grotscel afirman que es muy difícil predecir que tan difícil será una instancia particular del TSP-TW. Como ejemplo (y paradójica) de lo anterior muestran una instancia de 127 nodos que ocupó 3:49.82 horas de tiempo computacional, mientras que una de 43 nodos no pudo resolverse en menos de 5 horas. Esta instancia en particular de 43 nodos, resultó bastante difícil ya que en 5 horas de ejecución computacional, apenas se alcanzó un porcentaje de optimalidad relativa del 90.84% con más de 109 mil ejecuciones de programación lineal (llamadas al método “Simplex”) y más de un millón de hiperplanos generados por el algoritmo BC. Ascheuer, Fischetti, y Grotscel, en base a las experiencias computacionales, concluyen que las instancias del ATSP-TW en el límite de hasta 50 o 70 nodos, pueden ser resueltas hasta la solución óptima por el algoritmo BC.

A continuación en la Tabla 6.1, se presentan los resultados de dicho experimento. La segunda columna de la tabla corresponde al número de nodos que contienen cada una de las instancias que fueron aplicadas en el experimento. La última columna de la tabla corresponde a la cantidad de minutos computacionales que fueron requeridos en el experimento para que el algoritmo BC, lograra resolver cada una de las instancias respectivamente. Como puede ser comprobado en la última columna de la tabla, es posible darse cuenta que existen varias instancias que fueron utilizadas en el experimento, para las cuales no fue posible converger en una solución óptima aún después de 5 horas de ejecución computacional.

¹²¹ Ascheuer, N; Fischetti, M; Grotscel, M. (2001), Solving ATSP with time windows by branch-and-cut , Springer-Verlag, Germany, pag 297-299.

Finalmente, es importante precisar que la variante de nuestro proyecto de investigación referida a la condición de tener que considerar ventanas de tiempo negociables (restricciones de ventana de tiempo suaves) para cada cliente, es una variante relativamente poco explorada en el ámbito de la investigación bibliográfica. Tenemos como caso más reciente el trabajo conjunto de Pesant, Gendreau, Potvin, y Rousseau desarrollado en 1999. Es importante precisar que el trabajo de ellos proviene de una rama de la investigación completamente diferente a las que hemos venido revisando hasta ahora. Es decir no se trata de una metodología basada en BC ni tampoco en programación dinámica o metaheurísticas. Se trata de un enfoque relativamente nuevo y de la corriente europea denominada Programación Restrictiva (Constraint Programming, CP).

No obstante, es probado mencionar que CP ha tenido dificultades para dar soluciones competitivas respecto a sus contrincantes (por ejemplo el BC). El trabajo de Focacci y Milano prueba lo anterior. Ellos en el 2000, propusieron una metodología híbrida basada en el CP y en el BC para resolver el problema básico del TSP. Al final ellos concluyen que aunque su metodología superaba por mucho las soluciones ofrecidas por otras investigaciones basadas en implementaciones puras del CP, no obstante seguían estando lejos de ser comparables respecto a la calidad de las soluciones ofrecidas por los robustos esquemas de corte basados en el BC¹²³. En realidad la contribución de ellos fue proponer un esquema basado en el CP el cual ofrecía como ventaja la flexibilidad de que con pocas modificaciones, el mismo modelo básico podría dar tratamiento a la consideración de ventanas de horario simple y a la vez ventanas de horario múltiples ofreciendo de esta manera la posibilidad de poder negociar con el cliente.

A continuación en la Tabla 6.2, se resume la revisión bibliográfica que se ha venido exponiendo en los apartados 6.5 al 6.7 respecto a los estudios de investigación previos realizados para atender la solución computacional de problemas similares al nuestro. Con dicha tabla resumen damos por concluida la exposición de los proyectos de investigación previos desarrollados en el tema.

¹²³ Focacci, Filippo; Milano, Michela. (2000), Solving TSP with Time Windows with Constraints, Dip. Ingegneria, University of Ferrara, Italia, pag 4.

Tabla 6.2 Resumen comparativo de los estudios de investigación previos.

Investigador	Año	Taxonomía del Problema	Algoritmo	Dimensión	Ventanas de Horario	Desempeño Computacional
Baker	1988	ATSP-TW	BB	= 50 nodos	Cerradas, Poco Traslape	
Dumas, Desrosiers y Soumis	1993	PDP	Flujo de Redes			
Bruggen, Lenstra y Schuur	1993	PDP-TW	Herrística Lin-Kernigham y Enfriamiento Simulado	= 50 nodos	Cerradas, Poco Traslape	Instancias fuera de estas condiciones requerían = 10 Mins
Applegate y Bixby	1994	TSP-TW	BB			
Van Eijl	1995	VRP	BB	= 15 nodos	Abiertas	
Coth y Vigo	1995	PDP-TW	Meta-Heurística Tabu Search	50 = n = 100 nodos		Instancias para n = 100 nodos requerían = 1 Hora
Dumas y Solomon	1995	VRP-TW	Método de Descomposición de Dantzig-Wolfe	100 = n = 200 nodos	Cerradas, Poco Traslape	Instancias para 100 = n = 200: 20 = mins = 30
Balas y Simonetti	1996	TSP-TW	Programación Dinámica	40 = n = 50 nodos	Cerradas, Poco Traslape	= 2 Horas
Mingozi, Bianco, y Ricciardelli	1997	TSP-TW	Híbrido: BB y Programación Dinámica	= 120 nodos		
Ascheuer, Jünger y Reinelt	2000	ATSP-TW	BB con generación de hiperplanos	50 = n = 70 nodos	Abiertas y con bajo % de incidencia de TW	10 = mins = 20
Ascheuer, Fischetti, y Grottschel	2001	ATSP-TW	Híbrido: BB con Heurísticas de Intercambio	70 = n = 233 nodos	Cerradas, Poco Traslape	5 mins = Tiempo = 5 hrs

6.8 Comentarios finales: delimitación y diferenciación de la variante del problema de investigación (SPDP-sTW).

Para las conclusiones del presente capítulo buscaremos a la vez cubrir los siguientes 4 objetivos:

- a. Resumir las propiedades específicas a ser consideradas en el problema de investigación.
- b. Delimitar el problema de investigación.
- c. Diferenciar el problema respecto a otros proyectos de investigación previos.
- d. Contribuir a ofrecer elementos para la justificación del proyecto de investigación

A continuación se enumeran las características relevantes de nuestro problema de investigación:

1. El algoritmo identificará aquellos clientes que por su cercanía geográfica convenga, en su caso, renegociar su ventana de horario. Este elemento resulta sumamente útil en lo referente a las aplicaciones de ruteo en la práctica de las empresas, ya que de esta manera se evita generar recorridos de ruta costosos sobre la base de un supuesto nivel de servicio en el cliente que potencialmente resulta mejor re-negociar.
2. De lo expuesto en el punto anterior, es requerido desarrollar un análisis para medir el nivel de incidencia con el cual es conveniente renegociar las ventanas de horario con los clientes y así entonces verificar el nivel de esfuerzo requerido por parte de la empresa y a la vez el nivel de apego a las ventanas de horario originalmente deseada por el cliente.
3. La mayor parte de las investigaciones en el PDP-TW se centran en aquellos casos en los que se tiene definida una sola actividad a realizar en cada nodo, es decir se recoge producto en un lugar y se entrega el producto en otro. En el caso del proyecto de investigación nuestro, nos estamos concentrando en aquella situación en la cual se llega con un cliente al cual se le tiene que entregar un pedido. Más aun a este mismo cliente se le tiene que recoger todo aquel producto que ya no vaya a ocupar.
4. La devolución de producto mencionado en el punto anterior, puede obedecer a varias razones tales como:
 - i. Producto defectuoso o fuera de la fecha de caducidad.
 - ii. Producto devuelto a solicitud del cliente.

iii. Empaque retornable (o también denominada Logística Inversa).

5. La mayor parte de las investigaciones en el VRP-TW se enfocan solo en asegurar que la capacidad de carga del vehículo utilizado, sea suficiente para albergar la suma total del producto que se va a entregar a largo de la ruta. Sin embargo, en nuestro problema ocurre que en cada uno de los clientes a ser visitados en la ruta, existe la posibilidad de entregar pero también de recoger producto, lo cual naturalmente ocupa un volumen de carga.
6. La circunstancia anterior origina que cada vez que se termine la operación con un cliente, sea necesario re-calcular el saldo del espacio disponible para cargar en el vehículo. Esto quiere decir que la restricción de capacidad de carga debe ser verificada dinámicamente a la salida de cada uno de los clientes a los cuales se va a llegar. La situación anterior invalida la posibilidad de manejar tan solo restricciones de agregación a nivel de ruta tal y como sucede en el caso del VRP-TW.
7. Los problemas de PDP-TW generalmente consideran una sola carga a poder ser atendida por evento. Es decir, desde el momento en que se carga el producto hasta el momento en que se descarga, el vehículo solo está sirviendo a un solo cliente. En nuestro problema de investigación sucede que mientras se ejecuta el proceso de carga y descarga, el contenido del vehículo está ofreciendo servicio al producto que a muchos clientes se les va a entregar o simultáneamente a muchos clientes se les acaba de recoger.
8. Las ventanas de horario que nuestro proyecto de investigación requiere manejar tanto para la entrega de producto como también para la recolección del mismo en el sitio especificado por el cliente, son normalmente mucho más amplias que lo que habitualmente se maneja en los casos típicos del problema PDP-TW.
9. La razón de lo anterior se explica primero debido a que en nuestra instancia particular a atender, los clientes normalmente están dispuestos a ser atendidos en períodos de tiempo muy amplios ya que en condiciones especiales existe la posibilidad de ocupar varias eventos de entrega de producto al día.

10. Por el otro lado, las instancias típicas del PDP-TW revisadas en la bibliografía se centran en aplicaciones muy orientadas al tratamiento de embarques y recibos programados sobre la base del manejo de citas ya muy acotadas en cuanto a la amplitud de tiempo en las cuales se están comprometiendo para su ejecución en la práctica operativa.
11. En la práctica de las empresas pueden llegar a existir diversos objetivos en consideración al momento de tener que enfrentar un problema relacionado al PDP-TW. En términos generales existen 3 objetivos principales:
 - a. Minimizar el tiempo total requerido para atender el total de los clientes de la ruta (esto equivale a minimizar el “makespan” en los problemas de programación de la producción).
 - b. Minimizar el costo total de los traslados requeridos para la atención de la ruta.
 - c. Minimizar la falta de nivel de servicio o insatisfacción que puedan tener los clientes por llegar a atenderlos fuera de la ventana de horario comprometida previamente.
12. Como puede verificarse, el primer objetivo tiene que ver más con un sentido de eficiencia de operación enfocado a que la ruta termine su jornada de trabajo lo antes posible. De acuerdo a la empírica del postulante, pocas veces este objetivo es una necesidad real en la práctica de las empresas de distribución en ruta lo cual desincentiva su experimentación en nuestro caso.
13. Más aún como puede verificarse en el segundo objetivo, es de esperarse que si se logra obtener un recorrido con un costo mínimo, también es de esperarse que se obtenga al menos un buen indicador en cuanto a la duración de las jornadas de trabajo de las rutas.
14. El tercer objetivo pudiera ser interesante explorar en aquellos casos en lo que sea una posibilidad el poder manejar ventanas de horario “flexibles”. Un ejemplo de lo anterior sería aquel caso en el que por cada cierta cantidad de minutos que se llegue fuera de la ventana de horario, se tendría que incurrir en una penalización (o un costo de oportunidad) por la aplicación de un descuento aplicado en el precio del servicio (o del producto) ofrecido al cliente.

15. Puede entonces verificarse que resulta innecesario utilizar el enfoque anterior en aquellos casos en los que se trate de ventanas de horario “negociadas” y mucho menos “fijas”, ya que en estos casos lo que se busca es dar tratamiento a las ventanas de horario a través de restricciones a cumplir y no de objetivos a lograr.
16. Así pues, decidimos emplear el segundo objetivo que es el de minimizar el costo total de los traslados requeridos para la atención de la ruta. El costo incurrido de los traslados en una ruta puede ser una función lineal de la distancia o del tiempo requerido para dicha operación.
17. En ocasiones resulta ser más útil relacionarlo con el tiempo, sobre todo cuando se trata de rutas con recorridos urbanos donde existen áreas con un tráfico intenso. Así por ejemplo, si una ruta gasta \$1000 por día en mano de obra, combustible y mantenimiento, pues entonces aquellos traslados entre clientes que ocupen mayor cantidad de tiempo también de manera proporcional tendrán una participación mayor en el costo.
18. Es importante puntualizar en que no es lo mismo hablar de una ventana de horario fija que de una ventana de horario cerrada. Así por ejemplo puede existir un cliente con una ventana de horario fija que este dispuesto a recibir su atención en el rango de las 10:00 hrs. hasta las 16:00 hrs. Esta ventana de horario se dice que es fija porque si no se cumple entonces ya no puede darse el servicio. Sin embargo también puede percibirse que la ventana de horario es bastante amplia.

Con todo lo expuesto en este apartado, no pretendemos establecer que nuestro problema de investigación (SPDP-sTW) sea más difícil o fácil de resolver que cualquier otra variante del PDP-TW o del VRP-TW. Simplemente se trata de un problema con una variante diferente la cual conviene ser atendida y de ahí que se haya establecido como objetivo al inicio del apartado el dar una justificación para la investigación del problema.

Así entonces con esta exposición bibliográfica, concluimos la revisión de todos aquellos trabajos de investigación que previamente se han desarrollado en lo referente al tratamiento de problemas similares a nuestro SPDP-sTW. Pasamos ahora en el siguiente capítulo 7, al planteamiento de los objetivos e hipótesis del proyecto de investigación.

7. Planteamiento de objetivos e hipótesis del proyecto de investigación.

7.1 Introducción.

En este momento, los capítulos ya revisados anteriormente, nos serán de utilidad al tener presentes todos los antecedentes que existen alrededor del problema de investigación ya planteado (SPDP-sTW). A continuación los enumeramos con la intención preparar el terreno para enunciar el objetivo de nuestro proyecto.

1. El problema planteado es de naturaleza combinatoria y tiene una complejidad documentalmente probada a través de proyectos de investigación en el mundo al estar catalogado como NP-Hard, lo cual lo hace matemáticamente muy difícil de resolver.
2. El problema de investigación planteado posee ciertas características específicas que lo hacen diferenciable de proyectos de investigación previos y lo sitúan como una variante digna de exploración. El sustento de lo anterior puede ser verificado en el apartado 6.8.
3. En la práctica de la Logística de distribución de las empresas es común encontrar este tipo de problemas. Particularmente podemos referenciar a las empresas que se dedican a la entrega y recolección en ruta de productos que alternativamente requieren incluir en su proceso la retornabilidad del producto. Ejemplos de este tipo de negocios, van desde la industria embotelladora de refrescos y cerveza, pasando por las botanas y el pan, continuando con los productos lácteos y los jugos y terminando con los dulces y los cigarros.
4. Las instancias del problema de investigación, comúnmente encontradas en la práctica de las empresas, oscilan en el rango de los 60 a los 100 clientes (nodos) por ruta al día.
5. En la operación diaria y tomando en consideración el volumen de rutas de distribución a tener que administrar, normalmente las empresas disponen de poco tiempo para tomar decisiones respecto a cómo dar solución al problema en cuestión. Según la empírica del postulante, se

estima que 5 minutos pueda ser un tiempo que resulte práctico y útil aplicar como límite superior para el procesamiento correspondiente a la ruta de un vehículo.

6. Los algoritmos de solución exacta para solucionar problemas como el nuestro, consumen una cantidad de tiempo computacional que excede por mucho los 5 minutos que se acaban de definir en el punto anterior. Por tanto, b anterior limita a los algoritmos de solución exacta en su aplicación en la práctica de las empresas debido a que estas últimas requieren soluciones en tiempo real. El sustento de lo anterior puede ser revisado en el apartado 6.1.
7. Lo anterior hace requerir soluciones que aunque no sean matemáticamente óptimas, si sean razonablemente buenas ($> 90\%$ del óptimo) y que puedan ser obtenidas en poco tiempo computacional (= 5 minutos).
8. Un porcentaje bajo (7 de 55) de las empresas muestreadas del área metropolitana de la ciudad de Monterrey (AMM) han intentado resolver este tipo de problema a través de la implementación de sistemas APS's. Las empresas del AMM no han tenido éxito con este tipo de sistemas. El sustento de lo anterior puede ser revisado en el estudio de campo desarrollado en el apartado 3.8.
9. Las empresas del AMM, en términos generales administran el proceso de la logística de manera manual y en base a la experiencia de la gente, lo cual limita la posibilidad de obtener mejores soluciones que logren generar beneficios en términos de disminución de costos. El sustento de ello también puede ser corroborado en el apartado 3.8.
10. En las empresas del AMM, es muy poco frecuente (menos del 6%) que los profesionistas lleguen a aplicar las matemáticas en la toma de decisiones en general ni tampoco para solucionar el problema de la logística de distribución en lo particular. El sustento puede ser revisado en el estudio de campo desarrollado en el apartado 2.5. De lo anterior, el postulante opina que dicha falta de aplicación de las matemáticas por parte de los profesionistas, inhabilita a su vez de modo indudable, el que haya la posibilidad de una mejora en la generación de soluciones para atender la problemática de la logística en la práctica.

7.2 Planteamiento del objetivo central del proyecto de investigación.

De todo lo antes expuesto, entonces surge la pregunta:

¿Cómo podemos encontrar soluciones razonablemente buenas y en un tiempo computacional práctico, para resolver el problema de Ruteo de distribución para entrega y recolección de producto con restricciones de ventana de horario negociables (SPDP-sTW) ?

De lo anterior, enunciamos ahora el Objetivo Central del Proyecto :

Determinar mediante un experimento de laboratorio si el Algoritmo propuesto e instrumentado a través de la aplicación de “Algoritmos Genéticos”, resulta ser viable para encontrar soluciones razonablemente buenas (arriba del 90% optimalidad) y en poco tiempo computacional (menos de 5 minutos) para resolver el problema de Ruteo de distribución para entrega y recolección de producto, con restricciones de ventana de horario negociables (SPDP-sTW). El postulante estima que ambos parámetros (arriba del 90% de optimalidad y en menos de 5 minutos) pueden ser de utilidad práctica para ser aplicados en la logística.

7.3 Planteamiento de los objetivos específicos y entregables del proyecto de investigación.

A continuación se enumeran los objetivos específicos del proyecto de investigación.

1. Desarrollar un generador computacional de instancias aleatorias uniformemente distribuidas con la finalidad de aplicarlas para la instrumentación de la prueba experimental.
2. Generar las instancias numéricas de nuestro problema SPDP-sTW que vayan a requerirse para la prueba experimental (revisar el apartado 8.11).
3. Modelar matemáticamente hablando el problema del SPDP-sTW a través de una formulación basada en técnicas de programación mixta entera (MIP).
4. Desarrollar e implementar computacionalmente un algoritmo de solución exacta basado en las estrategias de “Ramificación y Corte” (BC) para aplicarlo a las instancias generadas en el punto No. 2 y obtener así el cálculo para la cota inferior para el modelo generado en el punto No. 3.
5. Desarrollar e implementar computacionalmente un algoritmo metaheurístico basado en las estrategias de “Algoritmos Genéticos” (AG) para aplicarlo a las mismas instancias generadas en el punto No. 2 y resueltas hasta la optimalidad en el punto No. 4.
6. Desarrollar un experimento cuya finalidad sea comparar las soluciones obtenidas con el Algoritmo Genético en el punto No. 6 versus la cota inferior obtenida en el punto No. 5.

Es importante mencionar que fue en el apartado 6.4, en donde se estableció la pertinencia de aplicar el algoritmo de “Ramificación y Corte” (BC) para calcular la cota inferior. Es importante mencionar que aunque técnicamente el algoritmo de solución exacta basado en la técnica de “Ramificación y Corte” (BC) está orientado en precisamente obtener la solución óptima para el problema, el objetivo de nuestro proyecto de investigación no está enfocado en demostrar la optimalidad de dicha solución. Como ya se definió antes, el propósito de lo anterior es que la solución obtenida mediante este método nos sirva como cota inferior para el análisis comparativo versus el Algoritmo Genético. A esta cota inferior, en términos estadísticos se le denomina “Grupo Control”.

7.4 Delimitación del proyecto de investigación.

El modelo matemático así como la propuesta algorítmica basada en la aplicación de un algoritmo genético, tienen el objetivo de reflejar los elementos que ocurren en el problema denominado “*Ruteo de distribución para un vehículo con entrega y recolección de producto con restricciones de ventana de horario negociables (SPDP-sTW)*”. Se busca integrar las consideraciones y supuestos más relevantes, matemáticamente hablando, acerca de cómo el fenómeno se presenta en la realidad. De esta manera el proyecto doctoral a la vez que experimental será explicativo, ya que documentará en el modelo matemático, la forma mediante la cual las variables endógenas guardan una relación causal con las exógenas.

Además de lo anterior, el modelo experimental tiene como finalidad encontrar una solución cercana al óptimo para el problema a resolver. De este modo establecemos que el modelo experimental no tan solo busca simular o predecir la realidad del fenómeno, sino explorar el espacio solución de una manera eficiente de tal forma que sea posible aproximar la solución óptima para el problema. Cuando nos referimos al calificativo de eficiencia nos estamos refiriendo a que se reúnan las siguientes propiedades:

1. Dimensionalidad (*cantidad de nodos en cada instancia*): $100 = w = 120$
2. Tiempos de ejecución computacional = *5 minutos*
3. Grado de calidad : *porcentaje de optimalidad > 90%*

La evidencia de “Viabilidad” del modelo propuesto será a través de la vía de la cuantificación matemática de su efectividad a ser evaluada durante el desarrollo del análisis experimental. Así pues la "viabilidad" será abordada de dos maneras:

1. Que el modelo logre abstraer matemáticamente el fenómeno que se presenta en el problema.
2. Que el modelo logre resolver eficientemente el problema.

En lo concerniente al producto entregable, éste será presentado en 3 modalidades:

1. En Lenguaje Natural: con la finalidad de explicarlo.
2. En Lenguaje Matemático: para brindar formalidad.
3. En Lenguaje Computacional: para implementarlo y probarlo .

Es importante precisar que la definición de las variables independientes y dependientes en el modelo propuesto no depende de pruebas de correlación estadística sino más bien de acuerdo a la empírica del

postulante y apoyado por la evidencia documental bibliográfica perteneciente a otras investigaciones que se hayan realizando en proyectos similares al problema del SPDP-sTW.

Aunque ya se había precisado en la exposición de los objetivos específicos en el apartado 7.3, que el presente proyecto de investigación no esta enfocado en probar la optimalidad de las soluciones arrojadas por los algoritmos propuestos, es importante mencionar que de haber sido ese el objetivo hubiera existido una dificultad probada para poder llevar a cabo un análisis comparativo entre la propuesta del proyecto de investigación versus las propuestas provenientes de investigaciones previas en el tema. La razón de lo anterior obedece a los siguientes argumentos:

1. La mayor parte de los proyectos de investigación provienen de una propuesta de aplicación para problemas específicos y muy particulares. Las restricciones impuestas en el problema suelen ser diferentes, los objetivos también pueden variar dependiendo de la empresa en donde se va a resolver el problema y más aún el significado de algunos conceptos, tales como el nivel de servicio al cliente, pueden llegar a ser diferentes en cada negocio. Como evidencia de lo anterior, se puede revisar la tabla 6.2, en la cual se muestra en forma comparativa, los estudios de investigación que fueron encontrados en la revisión bibliográfica.
2. Es fácil entonces comprender que la cantidad de problemas específicos resulta inmenso y es poco probable que al menos dos proyectos de investigación ataquen exactamente la misma variante del problema.

Finalmente en lo que se refiere a la problemática propia de la implementación en la práctica del modelo, este aspecto está limitado por la participación de los recursos económicos y humanos de las empresas, por lo que en este proyecto de investigación, la etapa de la implementación queda fuera del alcance comprometido. No obstante, las instancias numéricas que serán utilizadas para demostrar la viabilidad de los modelos, serán tomadas a partir de condiciones típicas de la realidad.

7.5 Justificación científica del problema de investigación.

De acuerdo a la revisión bibliográfica realizada, hemos procurado establecer las similitudes al igual que las diferencias que el planteamiento de nuestro problema guarda respecto a otros proyectos de investigación desarrollados previamente. Lo anterior no es trivial ya que precisamente lo anterior contribuirá a dar sustento y justificación a la investigación desde el punto de vista científico. Recordemos que en el apartado 6.8, la delimitación del problema de investigación contribuyó a evidenciar algunas características específicas que hacen diferenciable y a la vez justificable la variante de nuestro problema de investigación. No obstante creemos importante enumerar los siguientes argumentos adicionales para reforzar la justificación del proyecto:

1. El problema del PDP-TW apunta hacia una de las más ricas clases de problemas combinatorios en el área del OR/MS. El PDP-TW es un problema NP-Hard¹²⁴.
2. En la revisión bibliográfica del PDP-TW, hemos identificado algunos “huecos”; entre ellos descubrimos que la variante para la consideración de ventanas de horario negociables ha sido un tema de investigación muy poco explorado en términos relativos¹²⁵.
3. Adicionalmente, el problema del PDP-TW, en relación a sus áreas de aplicación, la variante menos explorada ha sido la relacionada al tema de la distribución física para entrega y recolección de producto y bienes de consumo¹²⁶.
4. La variante de nuestro problema de investigación SPDP-sTW posee como parte de su estructura, ventanas de horario relativamente anchas (arriba del 50% de apertura). Esta propiedad ocasiona por consecuencia que el problema tenga un espacio solución más grande.
5. El PDP-TW usualmente en la práctica, se trabaja con instancias de gran escala (instancias arriba de 70 clientes). Lo anterior hace prever la necesidad de investigar algoritmos de solución que puedan ofrecer soluciones razonablemente buenas en tiempos de ejecución

¹²⁴ Supra: revisar apartado 6.1

¹²⁵ Supra: revisar apartado 6.7

¹²⁶ Mitrovic, Snezana. (1998), Pickup and Delivery Problem with Time Windows “Technical Report SFU CMPT TR 1998-12”, Canada, pag 38-39.

computacional cortos ante la falta de tiempo para aplicar métodos de enumeración exhaustiva.

6. Se anticipa el requerimiento de un algoritmo que sea lo suficientemente “veloz” como para ofrecer soluciones aceptables en la práctica en un ambiente de toma de decisiones en línea.
7. Se requiere un algoritmo que aunque no ofrezca una solución matemáticamente óptima a costa de un tiempo computacional fuera del alcance práctico, si al menos ofrezca soluciones que sean razonables tanto en calidad como en el tiempo computacional requerido.
8. De acuerdo a la investigación bibliográfica que el postulante ha realizado, proveniente de proyectos de investigación previos en el mundo, es relevante mencionar que aunque los proyectos de investigación en el tema de Algoritmos Genéticos es amplísimo, no obstante es la primera ocasión en la que se va a experimentar con un algoritmo genético como propuesta de solución para el problema del SPDP-sTW en particular.
9. Los modelos y algoritmos propuestos por el postulante, buscan aprovechar las similitudes que nuestro problema tiene respecto a investigaciones hechas previamente. Las experiencias revisadas provenientes de otros trabajos de investigación nos hacen prever un comportamiento positivo para las iniciativas que serán propuestas.

No hay duda de que la familia de problemas del tipo PDP-TW son más difícil de resolver aún que el propio VRP-TW. La razón de lo anterior es puesto que el primero trata de una generalización del segundo ¹²⁷. Es decir, la familia de problemas pertenecientes al VRP, provienen de aplicaciones a problemas de naturaleza estática, mientras que los problemas de la familia del PDP son de naturaleza dinámica.

El razonamiento anterior justifica el dedicar el objetivo de la investigación no a probar la optimalidad de las soluciones ofrecidas por los algoritmos sino más bien enfocado en medir su efectividad sobre la base del costo-beneficio que resulta al ofrecer soluciones razonablemente aceptables para aplicaciones a ser implementadas en tiempo real donde hay poca disponibilidad de tiempo para ejercer la toma de

¹²⁷ Palmgren, Myra. (2001), A Column Generation Algorithm for the Log Truck Scheduling Problem, Department of Science and Technology (ITN), Linköping University, Norrköping Sweden, pag 3.

decisiones. Esto último es el fundamento que precisamente caracteriza a los ambientes de distribución dinámicos.

Más allá de los argumentos ya antes revisados, es innegable que desde el punto de vista científico este tipo de problema es fascinante por su complejidad matemática. Investigadores en los campos de las matemáticas, las ciencias computacionales y por supuesto del OR/MS han dedicado décadas en tratar de desarrollar algoritmos que resuelvan eficientemente el problema en cuestión¹²⁸.

¹²⁸ Lenstra, K. (1990), A Variable Depth Approach for the Single -Vehicle Pickup and Delivery Problem with Time Windows, "COSOR No. 90-48", Eindhoven University of Technology, Alemania, pag 35-90.

7.6 Relevancia práctica del problema de investigación.

En el capítulo 3 se evidenció que uno de los problemas más comunes en la práctica de la logística operativa de las empresas corresponde al problema de distribución de ruta. De acuerdo a la empírica del postulante, se estima que la mayor parte de las operaciones de entrega y recolección de producto en ruta en la práctica real de las empresas, están sujetas a restricciones de horario en el cliente y a restricciones de capacidad de carga. El proyecto de investigación se enfoca precisamente a este tipo de problemas (SPDP-sTW). A continuación se exponen algunas argumentaciones, las cuales tienen como objetivo dar a conocer la relevancia práctica del problema de investigación:

1. No es trivial que en las empresas que enfrentan los problemas de logística de ruteo, actualmente requieren de sistemas (o software) especializados con la finalidad de obtener rutas económicas para la distribución de sus productos. No obstante, la oferta computacional (sistemas APS) para resolver este tipo de problemas es cara y poco rigurosa.
2. Las empresas tales como las muestreadas en el antecedente del capítulo 3, descansan este tipo de decisiones sobre una base dependiente de la experiencia de la gente que viene ejecutando esta operación de manera empírica. Lo anterior limita la posibilidad de obtener mejores soluciones que logren generar beneficios en términos de disminución de costos.
3. Resultan evidentes los costos asociados en la solución óptima del problema para los cuales apenas se consiga un ahorro marginal en su ejecución, pueden llegar a resultar en ahorros sustanciales en términos absolutos para la operación del negocio ¹²⁹. Por tanto, se justifica el interés para dedicar investigación en este problema y conseguir estos ahorros potenciales en el ámbito de la distribución física de productos.
4. Aunque la optimización de los costos en la distribución física es un objetivo frecuentemente buscado en las empresas, no obstante existen otros beneficios que no deben ser subestimados. Por ejemplo, en el planteamiento del problema se puede apreciar que la necesidad de tener que considerar una ventana de horario para la atención de cada uno de los clientes, representa una fuerte necesidad y compromiso con el nivel de servicio que la empresa esté dispuesta a ofrecer.

¹²⁹ Savelsbergh, M. (1998), Local Search in Physical Distribution Management, Eindhoven University of Technology, Rotterdam Holanda, pag 5.

7.7 Propuesta metodológica para la solución del problema de investigación.

El presente apartado tiene el objetivo de exponer los elementos desarrollados en la propuesta de solución para el problema de investigación. Dicha propuesta de solución está compuesta por 4 rutinas de pre-procesamiento y otra de post-procesamiento en referencia a donde aparece la rutina del Algoritmo Genético Generacional. A continuación se enumeran las 6 fases:

1. Fase de descomposición basado en el método SPP (problema de la ruta más corta o Shortest path problem en inglés) para el análisis y explotación topológica de la versión original de la red logística.
2. Fase de compresión vía estrategia de clusterización mediante criterios de maximización de afinidad.
3. Fase de compresión discriminante de arcos basada en la heurística de los “k” nodos vecinos más cercanos para obtener la versión compacta de la red.
4. Fase de generación agresiva de cortes aplicada a la versión compacta de la red.
5. Fase evolutiva generacional para explotar el “pool” de cortes generados en el paso 4.
6. Fase de descompresión para desagregar la ruta propuesta para la versión original de la red.

Es importante precisar que los parámetros utilizados para el funcionamiento de los 6 algoritmos, han sido aplicados en función a la empírica del investigador. Esta fuera del alcance del objetivo del proyecto de investigación, el determinar como los valores de los parámetros empleados en estos algoritmos puede llegar a modificar el comportamiento en el desempeño de los resultados obtenidos. El objetivo del proyecto de investigación es medir y contrastar el efecto producido en el % de optimalidad obtenido mediante esta metodología con respecto al grupo control al hacer variar el recurso computacional aplicado. A continuación se presentará de manera general los aspectos técnicos que describen a los algoritmos implementados en las seis (6) fases de la propuesta de solución.

Fase # 1: Fase de descomposición basada en el método SPP para el análisis y explotación topológica de la versión original de la red logística:

No hay duda que la complejidad topológica de los problemas de ruteo logístico está presente en los casos de la realidad. En ese sentido, el generador de instancias aleatorias desarrollado como parte del proyecto de investigación procuró simular dicha característica con la finalidad de apegar las instancias de investigación a los casos de la realidad. Dicha realidad en lo tocante a los aspectos topológicos del tránsito de una ciudad, tiene que ver con lo siguiente:

- a. Proporción de calles (arcos topológicos) con tránsito en ambos sentidos.
- b. Proporción de calles de un solo sentido (oriente, poniente, norte y sur).
- c. Proporción de calles sin salida.

Las instancias aleatorias serán generadas sobre la base de un 30% de incidencia aplicado al inciso “a”. Otro 60% para el inciso “b”, del cual 15% de incidencia corresponde a cada una de las 4 orientaciones cardinales. Y finalmente tenemos un 10% de incidencia aplicado para las calles sin salida.

Con lo anterior expuesto, es fácil darse cuenta que cada una de las esquinas incluidas en una zona a ser analizada, corresponde a un nodo topológico (virtual) a ser considerado como parte en la red. Se dice entonces que dicho nodo resulta ser virtual debido a que no es un nodo al cual se requiera llegar por razones de entrega y/o de recolección de los pedidos a ser atendidos en los clientes tangibles de una ruta. Luego entonces, la única razón por la cual dichos nodos virtuales requieren ser considerados como parte de la red topológica, es debido a que estos son requeridos como puntos de conexión para poder realizar un ruteo que resulte factible de acuerdo a las restricciones de tránsito en la zona a ser analizada.

En base a la empírica del investigador, podemos establecer que para un caso típico, una zona lo suficientemente amplia como para cubrir un total de al menos 100 clientes requeriría al menos un total de 50 manzanas, es decir 2 clientes por manzana. Luego entonces, si la definición topológica de cada manzana ocupa típicamente 4 nodos, entonces podemos definir que para un problema típico requerimos considerar alrededor de 200 nodos virtuales más adicionalmente otros 100 nodos que corresponden a los clientes reales a ser atendidos, es decir 300 nodos en total.

Con todo lo antes expuesto, el algoritmo requerido para esta primera fase de pre-procesamiento, requiere cubrir los siguientes 2 objetivos:

- a. Identificar la secuencia de movimientos requeridos en términos cardinales (norte, sur, oriente, y poniente) para trasladarse óptimamente entre cada uno de los clientes (nodos reales).
- b. Calcular el costo incurrido en dicho movimiento.

Así entonces, el investigador hará uso de un algoritmo basado en el SPP con la finalidad de cubrir ambos objetivos arriba mencionados. Dicho algoritmo SPP es aplicado de manera combinatoria sobre los 100 clientes (o nodos reales) que estén siendo considerados en cada instancia aleatoria a ser revisada en el experimento. El resultado de dicho procedimiento combinatorio es entonces vaciado en dos matrices independientes de magnitud (100 x 100). Los 10 mil elementos de cada matriz hacen referencia a la logística requerida para ir del cliente “i” al “j”. La primera matriz está dedicada al cálculo del costo y la segunda matriz se enfoca en identificar la secuencia de movimientos requeridos.

En cuanto a la complejidad matemática del SPP, es probado que dicho problema NO es NP-Hard, ya que su tiempo de ejecución computacional crece de manera polinomial respecto al tamaño del problema. La experiencia computacional del investigador al aplicar el algoritmo del SPP para las instancias antes mencionadas (300 nodos), es que en menos de 70 segundos de ejecución computacional se tiene completamente resuelta la totalidad de los 10 mil elementos de las dos matrices ya antes mencionadas.

No hay duda que la contribución final de esta fase de pre-procesamiento de la red, es la de simplificar el tratamiento de la versión original de la red. Dicho de otro modo, este pre-procesamiento permite pasar de un total de 300 nodos a tan solo 100 nodos. Lo anterior reduce sensiblemente la cantidad de nodos y a la vez el espacio solución para el problema en cuestión lo cual será crucial para las siguientes fases del método propuesto.

En lo referente a la implementación computacional del presente algoritmo, su codificación puede ser revisada en el Anexo A del documento de tesis.

Fase # 2: Fase de compresión vía estrategia de clusterización mediante criterios de maximización de afinidad:

Los métodos de compresión de redes tienen en general la finalidad de agrupar (o clusterizar) “ n ” nodos de una red para obtener como resultado una cantidad más reducida de “ m ” meta-nodos (*donde: $m < n$*). El objetivo de lo anterior es que los “ m ” meta-nodos resultantes, puedan contribuir a generar una red mucho más sencilla para su tratamiento durante las etapas posteriores de optimización.

Habitualmente una red simplificada que resulta al aplicar este tipo algoritmos, se rige por criterios únicamente de contigüidad geográfica. Es decir el algoritmo intenta agrupar a los clientes sobre la base de la vecindad. No obstante lo anterior, nuestro problema de investigación no puede ser atendido tan solo en función a este único criterio. Las estructuras de las ventanas de horario que están asociadas a cada uno de los “ n ” nodos obtenidos como resultado de la fase # 1, requieren ser compatibles entre si para poder agruparse. Dicho de otro modo, aunque un nodo resulte tener afinidad geográfica con respecto a otro, si acaso estos dos nodos no comparten ventanas de horario “similares”, entonces el algoritmo no los agrupará para formar un solo meta-nodo.

Ahora bien, la “similitud” tanto geográfica como en lo referente a las ventanas de horario requeridas para que dos o más nodos se agrupen, va a depender fuertemente del factor de compresión que el investigador decida ejercer sobre la red original. A mayor factor de compresión se aplique como parámetro en el algoritmo, menor similitud se requiere y viceversa.

En cuanto a la generación de los meta-nodos, existen diversos enfoques que pueden ser aplicados y todos ellos dependiendo del atributo que este siendo analizado. Por ejemplo en el atributo geográfico, podemos promediar las coordenadas geodésicas (x , y) de los nodos a ser agrupados. En lo referente a las estructuras de las ventanas de horario, es posible que el meta-nodo tenga una ventana de horario que sea suficientemente amplia como para incluir a todos los nodos que estén siendo agrupados.

La experiencia computacional del investigador en las instancias probadas, verifica que reducir una red de 100 nodos con un factor de compresión del 50% resulta ser convenientemente suficiente para esperar un beneficio en el tratamiento computacional de la red resultante para las fases posteriores de optimización, pero sin comprometer la “pureza” de la red en la conformación de los meta-nodos. La codificación computacional del algoritmo puede ser revisado en el Anexo B del documento de tesis.

Fase # 3: Fase de compresión discriminante de arcos basada en la heurística de los “k” nodos vecinos más cercanos para obtener la versión compacta de la red:

La aplicación del método discriminante de arcos tiene el objetivo de eliminar todos aquellos arcos entre los nodos que resulten poco probables de poder ser utilizados en la solución de la ruta. Así entonces, aplicar este tipo de heurística de manera previa (pre-proceso) a cualquier tipo de algoritmo combinatorio puede contribuir significativamente a disminuir el recurso computacional necesario para obtener la solución a un problema.

Una heurística para discriminar aquellos arcos poco probables a ser utilizados en la solución de un problema de redes, es a través de la aplicación de criterios de vecindad. Es decir, para cada nodo “X” en cuestión se prohíbe el uso de aquellos arcos que apunten a los nodos “Yi” que no logren estar lo suficientemente cerca del nodo “X”. Un método de prohibición que puede ser aplicado es a través del uso de valores suficientemente diferenciados (valores grandes) en los coeficientes de la función de costo. Lo anterior asegura que la función objetivo del modelo matemático incorpore suficientes elementos informativos para penalizar el espacio solución del problema. De lo explicado anteriormente, es fácil comprender que a pesar de que la función objetivo se robustece, no obstante el total del espacio solución no logra disminuirse, ya que la cantidad de arcos válidos en la red, y por consecuencia de variables binarias, sigue estando inalterado. Por el motivo anterior, más que una estrategia de penalización lo que el investigador aquí propone, es un método de exclusión.

En el caso nuestro, el método que el investigador empleó está basado en la heurística de los “k” nodos vecinos más cercanos, donde “k” representa ser un subconjunto de los “n” nodos reales de la red. Como es previsible tenemos que típicamente el valor de “k” debe ser un valor muy inferior al valor de “n”, ($k \ll n$). Entre menor sea el valor de “k” logramos que el procedimiento combinatorio sea menos demandante de recurso computacional, pero en contraposición, se incrementa la probabilidad de dejar fuera del espacio solución la solución óptima para el problema en cuestión.

En base a la experiencia computacional del investigador, aplicar un 20% como factor discriminante resulta ser suficientemente robusto como para asegurar que el espacio solución del problema modificado no esté comprometido. De esta manera la cantidad combinatoria de arcos válidos se reduce drásticamente, por ejemplo a un 80% de la cantidad de arcos contenidos en el problema original. La codificación del algoritmo puede ser revisado en el Anexo C.

Fase # 4: Fase de generación agresiva de cortes aplicada a la versión compacta de la red:

Esta fase del método algorítmico resulta ser crucial, ya que es la que tiene el objetivo de encontrar lo más rápidamente posible una solución factible (no necesariamente la óptima) al problema que esté siendo tratado. Como ya se trató ampliamente a lo largo del capítulo 6 del documento, la principal intratabilidad que nuestro problema SPDP-sTW posee es en lo referente a las restricciones de las ventanas de horario de los nodos así como en la capacidad de carga del vehículo. Ambas restricciones se ven confrontadas por la actividad de entrega y recolección de producto a lo largo de la ruta.

Así entonces, encontrar para el problema una solución al menos factible lo antes posible en términos del tiempo del recurso computacional resultará ser una estrategia útil para ser explotada en el siguiente paso correspondiente a la fase evolutiva. Se habla de una generación agresiva de cortes, debido a que el algoritmo va agregando cortes al espacio solución del problema tomando en consideración lo siguiente:

1. Las estructuras de las ventanas de horario de los nodos y
2. La capacidad de carga disponible en el camión en cada punto de entrega y/o recolección.

Ambas consideraciones son utilizadas para ir agregando cortes en la función de desempeño del algoritmo genético (función fitness). En un sentido topológico, estos cortes tienen el objetivo de delimitar en forma deliberada las secuencias de visita hacia y desde cada uno de los nodos que van apareciendo en las soluciones tentativas de ruteo para el problema. Así entonces, en aquellas soluciones tentativas en las que se verifique que se esté incumpliendo con al menos una de las dos restricciones mencionadas en el párrafo anterior, es entonces que se van a ir agregando los cortes necesarios para que vuelvan a seguir apareciendo en iteraciones posteriores.

Se dice que la generación de cortes es agresiva, debido a que la primera solución factible que se obtiene al ir aplicando de manera iterativa estos cortes, no necesariamente es la óptima. Dicho de otro modo, los cortes agregados al espacio solución del problema en las primeras iteraciones, pueden luego estar en contraposición a los cortes agregados en iteraciones posteriores. Esto puede causar que algunos cortes que hayan sido agregados al “pool” eventualmente estén en el mejor de los casos como redundantes y en el peor de los casos como innecesarios. Mantener en el “pool” del sistema de restricciones un corte innecesariamente equivale a estar en riesgo de dejar fuera del espacio solución algunas soluciones potenciales y mejores que la que actualmente se tiene. El algoritmo concluye una vez que se obtiene la primera solución factible para el problema.

Como ya se mencionó antes, en la presente fase algorítmica se agregan dos tipos de cortes. El primer corte tiene que ver con las estructuras de las ventanas de horario y el segundo con las restricciones referentes a la capacidad de carga del camión.

La lógica que se utiliza para generar los cortes que dan tratamiento a las ventanas de horario, inicia al identificar el nodo que tiene la mayor desviación respecto a su estructura particular de horario. Dicho de otro modo, una vez que se tiene un tour completo (ó factible), se verifica para cada nodo si éste cumple o no, con la estructura de horario impuesto. Si el nodo cubre la consideración entonces se prosigue a verificar el siguiente nodo incluido en el tour. En caso de que el nodo en cuestión, esté siendo arribado en el tour en un horario que esté fuera de la ventana de tiempo especificado, entonces quiere decir que el nodo no está siendo atendido correctamente. En tal caso se calcula la desviación en la estructura de horario para el nodo. Dicha desviación se calcula de la siguiente manera:

1. Si el tiempo en el cual se arriba al nodo es antes de la hora de apertura del nodo, entonces la desviación es igual a la diferencia entre el tiempo de apertura del nodo menos el tiempo en el cual se arriba al nodo.
2. En caso de que el tiempo en el cual se arriba al nodo es después de la hora de cierre del nodo, entonces la desviación es igual a la diferencia entre el tiempo en el cual se arriba al nodo menos el tiempo de cierre del nodo.

El procedimiento anterior se aplica para cada uno de los nodos del tour. Hecho lo anterior, se procede a identificar el nodo con la mayor desviación en el tour respecto a la ventana de horario, dicho nodo lo llamaremos “nodo pivote”. A continuación se verifican uno a uno todos los nodos del tour en los cuales su tiempo de arribo coincide con la estructura de horario del nodo pivote, a estos nodos se les denominarán “nodos afines”. Finalmente el algoritmo genera el corte a través de agregar una restricción que asegure que el “nodo pivote” se conecte a uno de los “nodos afines” a la estructura de horario. La generación del corte se representa matemáticamente de la siguiente manera:

$$\begin{aligned}
 & \text{sea } I = \{1..n\} \text{ (nodos de la red)} \\
 & k \in I \text{ (nodo pivote)} \\
 & j \in I = \{1..m\} \text{ (nodos afines a } k) \\
 & \sum_{j=1}^m (x_{jk} + x_{kj}) \leq 1 \quad \forall k \in I
 \end{aligned}$$

La restricción anterior, asegura que el nodo pivote “ k ” utilice al menos uno de los arcos que lo conectan a cualquiera de los nodos afines “ j ”. Los cortes generados mediante esta estrategia agresiva, se guardan convenientemente en un pool de restricciones separados del resto de las restricciones del sistema global. El procedimiento antes descrito se aplica de manera iterativa hasta que el recurso computacional encuentra la primera solución factible al problema. Dicha solución factible tendría que ser con un solo tour completo y respetando las estructuras de las ventanas de horario.

A partir de que se obtiene la solución factible anterior, entonces ya solo quedan por considerar las restricciones de capacidad de carga. Para esto se vuelve a aplicar el mismo procedimiento de cortes agresivos ya antes expuesto. Las diferencias en esta otra ocasión serán:

1. El criterio de desviación para identificar el nodo pivote ya no será el de las ventanas de horario, sino será aquel nodo que mantenga el máximo exceso de carga requerida a lo largo del tour con respecto a la capacidad de carga del vehículo.
2. Los nodos afines para el nodo pivote, igualmente se identifican a partir de revisar aquellos nodos que pueden ser conectados hacia el nodo pivote sin que se genere alguna violación a la restricción de capacidad de carga del vehículo.

Una vez que el recurso computacional encuentra la primera solución factible para el problema global, entonces el algoritmo finaliza el procesamiento computacional de la fase de generación de cortes. Dicha solución factible global es aquella que corresponde a:

1. Un tour completo que cubra el total de los nodos de la red.
2. Que el tour asegure que a cada nodo se llegue dentro de la ventana de horario estipulada.
3. Que las operaciones de carga y descarga aplicadas al arribar a cada uno de los nodos que conformar el tour, no violen la capacidad de carga del vehículo.

La codificación computacional del algoritmo que se ha venido explicando a lo largo del presente apartado, puede ser revisada en el Anexo D.

Fase # 5: Fase evolutiva generacional para explotar el “pool” de cortes generados :

La fase evolutiva de la metodología es en la que se busca aproximar la solución óptima para la versión compacta de la red logística definida en la fase # 3. Como es de esperarse, la cantidad de cortes que llegan a acumularse en el “pool” del sistema de restricciones es significativo. La experiencia computacional del investigador indica que la cantidad de cortes que tienen que ser tratados en la presente fase evolutiva llega a estar en el orden de los 15 a los 40 cortes dependiendo de la instancia numérica que esté siendo analizada. Entre más intrincada y/o compleja sea la estructura de las restricciones del problema, mayor es la cantidad de cortes necesarios para lograr encontrar la primera solución factible para un problema.

Es fácil comprender que el proceso de aproximación a la solución óptima verdadera de la versión compacta de la red se logra al identificar cuales de los cortes incluidos en el “pool” son convenientemente necesarios eliminar del “pool”. Dicho proceso de eliminación de cortes no puede ser visto de manera individual en cada corte, ya que la presencia y/o la eliminación de un corte puede estar comprometiendo simultáneamente la presencia y/o la eliminación de otro o de otros cortes a la vez. Así entonces, identificar cuáles son los cortes que conviene eliminar en el pool, debe ser visto cómo un sub-problema de naturaleza combinatoria.

La circunstancia antes descrita, hace que el problema sea apropiado para ser atendido mediante el uso de estrategias de recombinación evolutiva a través del uso de codificación binaria. La codificación binaria permite naturalmente representar dos posibilidades, las cuales en nuestro caso significan la eliminación (0) o la presencia (1) del corte en el “pool” del sistema de restricciones. Si acaso ejemplificáramos una instancia con 30 cortes congregados en el “pool”, entonces estamos hablando de que se requieren de 30 genes para conformar la estructura genética del organismo que representa la solución potencial para el sub-problema del “pool” de cortes. La cantidad de combinaciones resultante de este sub-problema es igual a **30!** (30 factorial, es decir $2.65252859812191 \times 10^{+32}$).

Aunque el espacio solución del sub-problema anterior es enorme, no obstante resulta mucho menor si lo comparamos con respecto al dimensionamiento del problema correspondiente a la versión compacta de la red. Para efectos de ejemplificar el dimensionamiento de una instancia típica para este último caso pudiéramos estipular los siguientes parámetros:

- a. Cantidad de nodos requeridos a tratar en la versión original de la red = 100 nodos
- b. Cantidad de nodos resultantes de la fase de compresión vía clusterización = 50 nodos
- c. Cantidad de arcos resultantes de la fase de compresión discriminante (obtención de la versión compacta de la red) = 50 nodos x 20 vecinos = 1000 arcos
- d. Aplicando la propiedad de simetría de la red tendríamos entonces un total = $1000 / 2 = 500$ arcos diferentes.

Luego entonces, si aplicáramos equivalentemente una codificación binaria para resolver este problema combinatorio, tendríamos entonces un total de combinaciones igual a **(500)!**. Nótese que el número más grande al cual se le puede calcular su función factorial en el paquete computacional de Microsoft Excel es **170!** (170 Factorial).

Como puede comprobarse, resolver nuestro sub-problema basado en la solución combinatoria del “pool” de cortes resulta ser extremadamente menos difícil en comparación a resolver el problema referido a la versión compacta de la red, ya no digamos si acaso se trata de solucionar el problema correspondiente a la versión original de la red.

En cuanto a la implementación del algoritmo evolutivo, la codificación binaria de los cortes se complementa con operadores de selección tipo torneo y con un factor de cruzamiento al 50%. Es decir, el material genético del organismo padre y del organismo madre es al 50% cada uno. El método de reproducción es a través de dos puntos de cruce generados aleatoriamente a lo largo de la longitud de la cadena cromosómica.

Por otro lado, el factor de mutación es inicializado a un 7.5% en la primera generación, pero se va auto-ajustando en cada generación subsecuente de acuerdo al cálculo porcentual de individuos que sean idénticos genéticamente hablando en la población. Dicho de otro modo, a medida que la población aumenta su nivel de degeneración, en una medida proporcional se aplica una curva exponencial de crecimiento en el porcentaje de mutación con un límite asintótico al 50%. Esto quiere decir que en el mayor de los casos, por cada dos nuevos organismos que nacen en la población, uno tendrá mutado su contenido genético en cierta posición cromosómica elegida de manera aleatoria. Finalmente el factor de elitismo está limitado a un 15% sobre el total de la población total.

La codificación del algoritmo aquí descrito puede ser revisada en el Anexo E.

Fase # 6: Fase de descompresión de la ruta propuesta para la versión original de la red:

La última fase de la metodología corresponde a la descompresión de la ruta a ser propuesta como solución para la versión original de la red. Aunque el proceso de optimización evolutivo se lleva a cabo durante la ejecución computacional de la fase # 5, no obstante, la solución obtenida corresponde para la versión compacta de la red y por tanto no resulta ser suficiente para dar respuesta al problema original en cuestión. De ahí que sea necesario incorporar en la metodología una última fase de post-procesamiento con la finalidad de ofrecer una solución que satisfaga los requerimientos iniciales.

Nuestra fase de post-procesamiento tiene el objetivo de propagar la solución de la red compacta en una solución que sea topológicamente equivalente a la versión original y a la vez asegure que las restricciones impuestas en la versión original del problema sean cubiertas. A las restricciones a las que nos estamos refiriendo son naturalmente a:

1. Las estructuras de las ventanas de horario de los nodos y
2. La capacidad de carga disponible en el camión en cada punto de entrega y/o recolección.

La presente fase contiene dos rutinas principales las cuales son:

1. Rutina de desagregación para convertir los meta-nodos formados en la fase # 2, en los nodos correspondientes a la versión original de la red. Por ejemplo para desagregar los 50 meta-nodos optimizados durante la fase # 5 evolutiva y convertirlos en los 100 nodos reales (clientes). Esta rutina de desagregación está enfocada en determinar la secuencia óptima en la cual los nodos que integran a los meta-nodos se deben desagregar. Por ejemplo, si acaso un meta-nodo “ x ” estuviera compuesto por 3 nodos y luego otro meta-nodo contiguo “ y ” tuviera 2 nodos, entonces el algoritmo debe establecer la secuencia óptima en la cual conviene ordenarse los 5 nodos entre sí dada la contigüidad de los dos meta-nodos “ x, y ”. Desde luego el problema no es tan sencillo, ya que por ejemplo en un caso como el nuestro, se tendrían que ordenar 50 meta-nodos considerando la secuencia obtenida en la fase # 5 evolutiva. No obstante lo anterior, la experiencia computacional del investigador muestra que el problema puede ser resuelto hasta la optimalidad en un tiempo computacional que no excede de los 15 segundos.

2. Rutina de ruteo topológico para convertir la red de nodos reales (clientes) en la versión original de la red. Debemos recordar que la versión original de la red puede llegar a contener típicamente 300 nodos, a continuación se ejemplifica lo anterior:
 - a. Cien (100) nodos reales (clientes).
 - b. Doscientos (200) nodos topológicos: a razón de 2 clientes por manzana, 100 clientes nos requeriría 50 manzanas. Si cada manzana ocupa típicamente 4 esquinas (nodos topológicos), entonces tenemos 200 nodos virtuales.

Básicamente lo que el algoritmo de ruteo topológico realiza, es establecer la secuencia de movimientos requeridos en términos cardinales (norte, sur, oriente y poniente) para trasladarse óptimamente entre cada uno de los clientes (nodos reales). Este proceso, debe tomar rigurosamente en cuenta la secuencia de visita establecida en la rutina de desagregación en donde los meta-nodos son convertidos de vuelta en los nodos reales (clientes) de la red.

Este proceso de reconversión descrito, hace uso de la información generada en la fase # 1. De esta manera, la rutina de ruteo topológico simplemente va sustituyendo las secuencias de visita entre los nodos reales (clientes) por las cadenas correspondientes a los movimientos cardinales que son necesarios para trasladarse de nodo en nodo. Dichas cadenas describen los movimientos en términos de dirección (oriente, poniente, norte y sur) que son óptimamente necesarios realizar en términos de los nodos topológicos (esquinas de las manzanas) que se requieren arribar para conectar a cada uno de los nodos reales (clientes) en la red. Igualmente, cada cadena de movimientos esta asociada al costo incurrido para hacerlo lo cual permite calcular implícitamente el costo de la ruta formada.

Con todo lo antes descrito, finalmente se tiene la solución propuesta para el ruteo de la versión original de la red. La codificación del algoritmo aquí expuesto puede ser revisada en el Anexo F del documento de tesis.

A continuación en la siguiente figura 7.1, se expone de manera esquemática el proceso algorítmico mediante el cual se resumen las 6 fases que fueron expuestas en el presente apartado y con lo cual el investigador da por concluida la explicación de la propuesta metodológica para el planteamiento del problema.

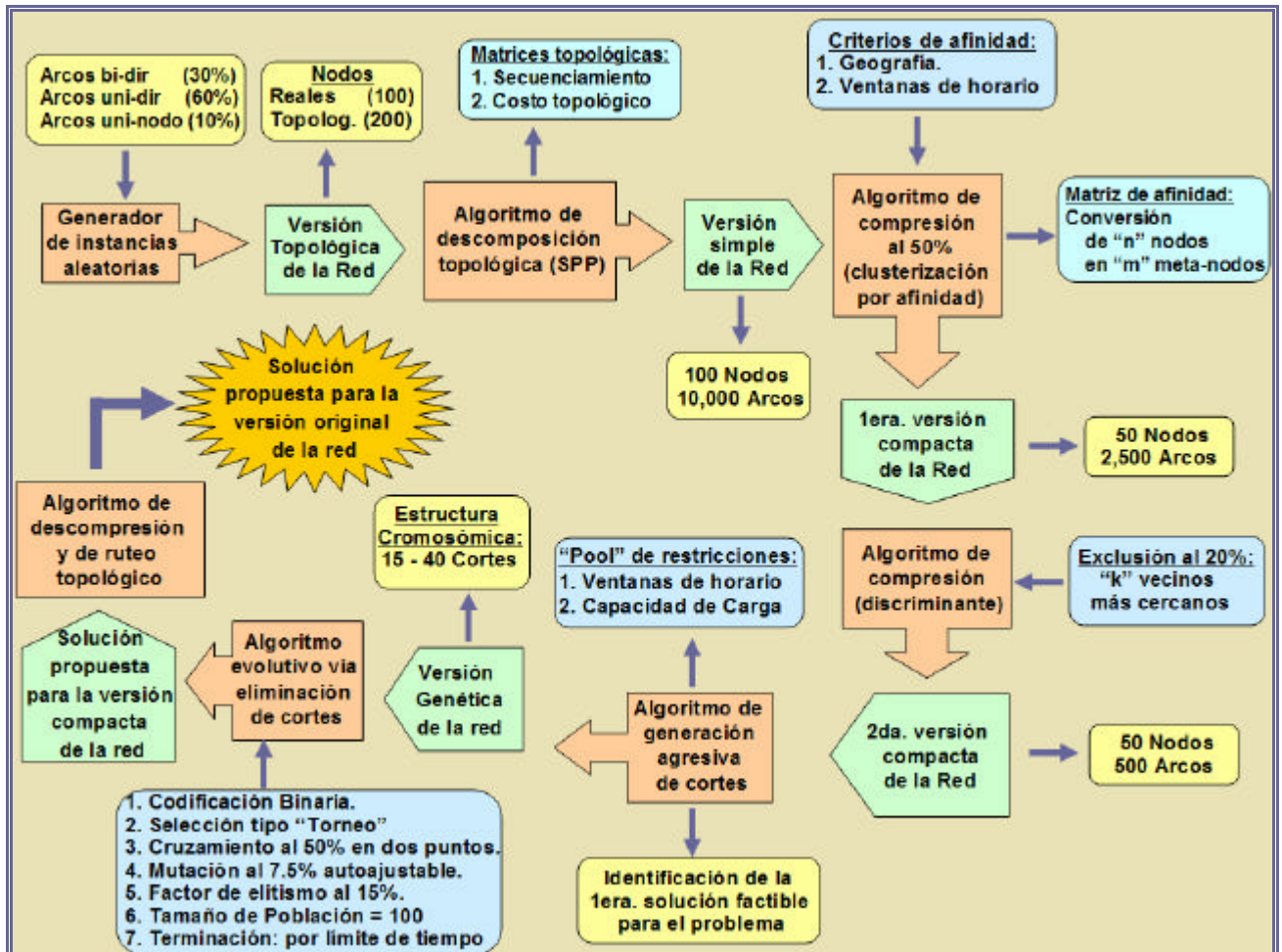


Figura 7.1 Diagrama de flujo para la propuesta de solución del SPDP-sTW.

7.8 Declaración de la hipótesis de trabajo del proyecto de investigación.

Con lo expuesto hasta este momento y tomando en cuenta la falta de evidencia teórica respecto a la aplicación exitosa de los algoritmos genéticos para la solución del problema SPDP-sTW, podemos establecer la siguiente hipótesis nula:

Ho: La metaheurística propuesta basada en la aplicación de un algoritmo genético para solucionar el problema de un vehículo único con entrega y recolección de producto con restricciones de capacidad de carga y con ventanas de horario negociables (SPDP-sTW), no logra mejorar el 90% de optimalidad en tiempos de ejecución computacional menores a los 5 minutos para instancias de tamaño $100 = w = 120$, (done w = cantidad de nodos en la red).

A continuación establecemos nuestra hipótesis de investigación:

Ha: La metaheurística propuesta basada en la aplicación de un algoritmo genético para solucionar el problema de un vehículo único con entrega y recolección de producto con restricciones de capacidad de carga y con ventanas de horario negociables (SPDP-sTW), antes de 5 minutos de esfuerzo computacional, logra mejorar el 90% de optimalidad respecto a la solución registrada en el grupo control y resuelto mediante un algoritmo exacto de Ramificación y Corte (BC) sin límite de tiempo, para instancias de tamaño $100 = w = 120$, (done w = cantidad de nodos en la red).

Resumiendo lo anterior, tenemos que:

Ho: $\mu_{\text{Grupo experimental}} \geq 90\% (\mu_{\text{Grupo Control}})$

Ha: $\mu_{\text{Grupo experimental}} < 90\% (\mu_{\text{Grupo Control}})$

Donde μ es la media poblacional correspondiente a los porcentajes de optimalidad alcanzados en la medición del 5to. minuto de esfuerzo computacional para el problema SPDP-sTW y para instancias de tamaño $100 = w = 120$, (done w = cantidad de nodos en la red).

7.9 Comentarios finales: taxonomía general del proyecto de investigación.

Resumimos al proyecto de investigación en términos de las siguientes propiedades:

1. Tipo de Investigación: explicativa y experimental.
2. Tipo de Hipótesis estadística: análisis de varianza mediante estadística no paramétrica.
3. Instrumento de Investigación:
 - i. Algoritmo de solución exacta basado en generación de constructiva del hiperplanos o cortes (BC).
 - ii. Algoritmo meta-heurístico de aproximación basado en un Algoritmo Genético de estado estable (AG).
4. Diseño de la Investigación: Experimental / Experimento Puro
5. Tipo de la Investigación:
 - i. Es descriptivo: ya que busca identificar los "que's".
 - ii. Es explicativo: ya que procura exponer los "como's".
 - iii. Es predictivo: ya que reacciona de manera predecible a las "causas y a los efectos" que caracterizan al fenómeno en cuestión dando a conocer los "Porque's".
 - iv. Y sobre todo es prescriptivo: ya que finalmente, el objetivo que busca alcanzar es el hallazgo de la solución al problema, es decir busca su condición de operación óptima.

Así pues, en este último sentido y siendo el problema de investigación del tipo "prescriptivo", entonces es claramente legítimo que la metodología propuesta para el "análisis de superficie" utilizado para el estudio analítico del espacio solución de un problema, no será realizado a través de la aplicación de las técnicas de regresión múltiple sino más bien a través de la aplicación de modelos de optimización matemática.

Concluido el planteamiento de los objetivos e hipótesis de la investigación, pasaremos en el siguiente capítulo a exponer el diseño experimental que será propuesto para dar respuesta a la hipótesis presentada.

Si tuviera 8 horas para derribar un árbol, destinaría 6 para afilar mi hacha.

Abraham Lincoln

8. Diseño experimental.

8.1 Introducción: objetivos del esquema general del experimento.

El objetivo principal de nuestro diseño experimental es el de evaluar el comportamiento del método propuesto versus el grupo control. Lo anterior se logra mediante un análisis comparativo que permita determinar que tan lejos o cerca la solución del Algoritmo Genético está de la solución óptima verificable a través del grupo control. Así pues, el análisis comparativo busca dar respuesta a lo siguiente:

1. ¿Cómo se comporta el modelo propuesto durante el proceso de búsqueda computacional ?
2. ¿Qué tan robusto es el modelo cuando éste está expuesto a consideraciones variadas ?
3. ¿Cuál es la probabilidad de que el modelo consistentemente encuentre o aproxime la solución óptima de un problema ?

Debemos partir de que el problema de investigación, será abordado en el experimento a través de dos instrumentos algorítmicos:

- a. Un algoritmo de solución exacta basado en el método de Ramificación y Corte (BC) vía la generación de hiperplanos. Este instrumento nos servirá para el desarrollo del grupo control del experimento.
- b. Un algoritmo de aproximación meta-heurístico basado en un Algoritmo Genético. Este instrumento será confrontado versus el grupo control en lo referente a diversos criterios de efectividad que serán tratados más adelante.

En los problemas de programación lineal (PL), sucede habitualmente que la aplicación del método “Simplex” puede ser visto como una “caja negra”. Es decir, no se requiere por parte del investigador hacer ningún tipo de manipulación en los parámetros de funcionamiento del algoritmo. Por otro lado en los modelos de programación mixta entera (MIP), siempre será necesario la intervención del investigador que contribuya a la manipulación de los parámetros del algoritmo que esté siendo utilizado para la solución del problema¹³⁰.

¹³⁰ Bixby, Robert. (1999), MIP: theory and practice closing the gap , Ilog Cplex Division, Department of Computational and Applied Mathematics Rice University, Houston EUA, pag 3.

Dada la naturaleza del problema de investigación (NP-Hard), es oportuno destacar que en ocasiones será muy difícil determinar la solución óptima exacta para una instancia particular del problema. Lo anterior es natural, debido a que nuestro problema de investigación tiene un tiempo de solución computacional que crece no en forma polinomial sino en forma exponencial. Con la explicación anterior, toma mayor relevancia aún el algoritmo de Ramificación y Corte (BC) ya que será éste el que defina al grupo control contra el cual se podrá medir el desempeño y la efectividad del Algoritmo Genético.

Así pues, el objetivo del algoritmo BC será el identificar la solución óptima verificable para cada instancia a ser aplicada como unidad experimental para el problema de investigación. Dicho de otro modo, la solución obtenida mediante el algoritmo BC será nuestra cota inferior para la prueba experimental a ser aplicada al Algoritmo Genético. Las cotas inferiores calculadas mediante la aplicación del algoritmo BC, aseguran una adecuada estimación del error experimental. Dicha estimación resulta ser una estrategia experimental bastante agresiva ya que garantiza obtener una medición precisa respecto al grado o efectividad con que el Algoritmo Genético está igualando el valor alcanzado por el grupo control identificado mediante el algoritmo de solución exacta basado en el BC. Lo anterior resulta favorable para el propósito de la validez del experimento del laboratorio.

El éxito del experimento no sólo depende de la forma en que se instrumente el Algoritmo Genético. El postulante está convencido de que la definición del grupo control jugará un papel crítico para la prueba de la hipótesis doctoral. Por tal razón, es plausible precisar que la contribución científica del experimento recae en la implementación de ambos instrumentos algorítmicos, es decir, tanto el Algoritmo de Ramificación y Corte (BC) así como el Algoritmo Genético.

El primer paso se da a partir de la implementación computacional del modelo matemático de programación mixta entera (MIP). Para el grupo control, éste se resolverá a través del algoritmo de planos de corte basado en la generación de hiperplanos vía el método BC. Para el desarrollo del algoritmo basado en el BC, nuestra propuesta será a través del estudio del poliedro que conforma el espacio solución para nuestro problema de investigación. Así entonces, el algoritmo propuesto, dinámicamente irá agregando hiperplanos (o cortes) a un sistema de restricciones del tipo $Ax \leq b$. La generación de los hiperplanos a ser manipulados en el experimento provienen naturalmente de la explotación de la estructura del problema asociado al polígono del espacio solución.

En general, las ideas desarrolladas en el campo científico en lo referente al método BC, han sido encaminadas hacia propósitos de investigación experimental. Hasta hace apenas 5 años, poco se había hecho en lo referente a su aplicación en el software de optimización para uso comercial. Como ejemplo de lo anterior podemos nombrar a LINGO©, OSL©, XPRESS-MP© y CPLEX© (marcas registradas), como aquellos que ya vienen incorporando mucho de estas capacidades y teorías acumuladas en el ámbito experimental ¹³¹.

Sobre la base de lo anterior, el algoritmo propuesto de investigación para la generación de los hiperplanos estará basado en un esquema de BC y se toma la decisión de implementarlo en el código computacional de XPRESS-MP© Versión (13.26). El postulante considera que el esquema de desarrollo algorítmico es robusto. A continuación en el apartado 8.2, se exponen los fundamentos del algoritmo basado en el BC para la conformación del Grupo Control.

¹³¹ Bixby, Robert. (1999), MIP: theory and practice closing the gap , Ilog Cplex Division, Department of Computational and Applied Mathematics Rice University, Houston EUA, pag 2.

8.2 Aplicación del procedimiento algorítmico de ramificación y corte (BC) como grupo control para la prueba experimental.

Un método trivial para resolver un problema de optimización combinatoria sería a través de una enumeración exhaustiva del espacio solución y seleccionando la mejor solución factible. No obstante el método anterior como ya se revisó antes puede consumir siglos o milenios en términos de recursos computacionales.

El método de “Ramificación y Corte” (o en inglés Branch & Cut, BC), fue desarrollado en 1960 por Land y Doig ¹³². Se trata de un algoritmo de enumeración parcial, el cual consiste en delimitar el espacio de búsqueda para la solución de un problema. Así pues, BC genera cortes (o restricciones) sobre la base de que cada uno de ellos delimite el espacio de búsqueda y por consecuencia logre incrementar la probabilidad de encontrar la solución óptima para el problema. Esta estrategia puede ser vista como la analogía al “divide y vencerás” en el que de una manera sistemática se van adicionando restricciones al problema original.

El algoritmo BC inicia resolviendo la versión relajada del problema original a atender¹³³. Lo anterior se logra simplemente mediante programación lineal. Si acaso la solución obtenida para el problema de programación lineal antes descrito logra satisfacer intrínsecamente las restricciones de integralidad, pues entonces decimos que el problema original ha sido resuelto. De otra manera quiere decir que es necesario agregar más restricciones (o cortes) que logren que el valor de las variables obtenidas mediante programación lineal no violen las restricciones de integralidad.

La condición anterior es lo que origina el proceso de ramificación. Es decir, a partir de cualquier variable x_j que no haya resultado entera (fraccional) se procede a ramificar dos sub-problemas descendientes del problema original. La primera ramificación buscará explorar el espacio solución correspondiente a un valor igual a cero, para esto se incluye la restricción (o corte) $x_j = 0$. La segunda ramificación buscará probar con el corte opuesto haciendo $x_j = 1$.

¹³² Land, A; Doig, H. (1960), An Automatic Method for Solving Discrete Programming Problems , *Econometrica*, EUA, pag 97-120.

¹³³ Supra: revisar pag 155.

A través de este procedimiento recursivo, se van desarrollando los cortes que van acotando el espacio solución obtenido mediante la relajación del problema original. Podemos resumir lo anterior mediante la siguiente Figura 8.1.

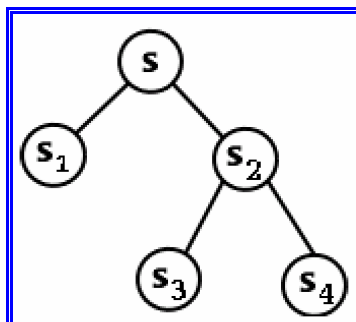


Figura 8.1 Proceso de bifurcación en el procedimiento de ramificación y corte.

Como puede ser revisado en la Figura 8.1, las bifurcaciones son generadas a partir de la adición de nuevas restricciones o cortes a partir del problema original S . Así entonces podemos definir lo siguiente:

$$S \rightarrow S^1 \rightarrow S^2; \text{ donde:}$$

$$S^1 \rightarrow S \rightarrow \{x \in R : x \leq x^*\} \text{ y}$$

$$S^2 \rightarrow S \rightarrow \{x \in R : x \leq x^*\}$$

Así pues, llevando al extremo el procedimiento anterior, esta sub-división de problemas pudiera parecer como un método de enumeración exhaustiva al cubrir todos los elementos en S . No obstante, si al procedimiento anterior le agregamos una rutina que permita revisar cuando ya no resulta necesario seguir bifurcando dentro de S_k , pues entonces lo que resulta es un procedimiento de enumeración parcial lo cual permite obtener ahorros en términos de recursos computacionales.

La explicación anterior resulta ser muy simple y general. Para ser un poco más formal definamos a P como el problema a tener que ser resuelto. Así entonces diremos que P tiene un conjunto de posibles soluciones factibles definidas por S . Entonces a lo largo de cualquier punto durante el proceso de búsqueda de la solución óptima para un problema, tendremos que S estará compuesta por un subconjunto S_0 de soluciones ya antes consideradas y como complemento otro subconjunto de soluciones aún por explorar S_1, S_2, \dots, S_n y así hasta agotar el total de soluciones factibles en S .

Matemáticamente hablando tenemos entonces que:

$$S = S_0 \cup S_1 \cup \dots \cup S_n$$

?

De acuerdo al procedimiento de búsqueda, cada uno de los S_k son mantenidos y actualizados en una lista de subproblemas candidatos P_k . A su vez para cada uno de los P_k se determina si acaso es necesario profundizar el proceso de búsqueda dentro del S_k correspondiente. Si la respuesta es negativa, entonces quiere decir que ya no hace falta profundizar dentro del sub-espacio solución representado por S_k y por tanto se procede a remover de la lista de problemas candidatos a P_k y se define a S_{k+1} como el siguiente espacio de solución a explorar. Si por el contrario, la respuesta es afirmativa entonces se aplica el proceso de enumeración al sub-espacio solución de S_k , lo cual se logra subdividiendo el sub-problema P_k en nuevos sub-problemas candidatos los cuales resultan al aplicar a S_k nuevos cortes y restricciones. El proceso antes descrito opera de una manera iterativa y recursiva hasta no tener ningún otro sub-problema que revisar de la lista de candidatos en cuyo caso se logra entonces determinar cual es la solución óptima.

El proceso mediante el cual se determina cuando un sub-problema P_k requiere subdividirse para profundizar la exploración de S_k se le denomina bifurcación. La bifurcación se desarrolla a partir de determinar para cada sub-problema P_k su límite superior U_k , su límite inferior L_k y su solución óptima definida por Z_k ; donde $L_k = Z_k = U_k$. En general, el proceso de bifurcación para un problema P_k no es necesario continuar cuando se haya verificado que el problema P_k no tiene soluciones factibles o cuando la solución óptima del sub-problema no resulte mejor que la última solución mejor encontrada hasta el momento a lo largo del proceso de búsqueda. A esta solución mejor encontrada se le denominada “titular” (o “incumbent” en inglés) y se le puede representar por I . Así pues, cuando $I = Z_k$ se identifica que ya no es necesario bifurcar en P_k .

Desde luego, si queremos aplicar lo anterior para un problema determinado, se requerirán dos cosas adicionales:

1. Un método adecuado para identificar en qué orden se irán revisando cada uno de los sub-problemas P_k . Lo anterior no es trivial, ya que entre más rápido se accedan los sub-problemas P_k que arrojen mejores valores óptimos Z_k , más rápido podrán ser desechados otros sub-

problemas P_k , lo que a su vez evitará tener que profundizar en sus sub-espacios S_k redundando en una acotación del espacio solución.

2. Un método adecuado para evaluar el límite superior S_k y el límite inferior L_k para cada sub-problema P_k . En general se busca lograr que los límites sean lo más acotados posibles, es decir que U_k y L_k estén lo más cerca entre sí. Entre más parecidos sean entre sí U_k y L_k , entonces se reduda en que la búsqueda de Z_k sea más rápida para converger. Tenemos entonces que el proceso de búsqueda de la solución óptima termina cuando:

$$U_k = L_k = Z_k$$

Los métodos utilizados para reducir el valor de U_k , han sido mediante heurísticas que logren encontrar lo antes posible las mejores soluciones enteras para el problema. Por su parte, los métodos utilizados para aumentar el valor de L_k , han sido mediante el desarrollo de algoritmos para la generación de planos de corte (o “cutting planes” en inglés) que logren limitar el espacio solución. A continuación resumimos el procedimiento BC a través del siguiente pseudo-código:

1: Seleccionar un sub-problema S_k y definirlo como activo.

2: Si S_k resulta infactible entonces

Borra S_k

Se vuelve al paso 1.

De otro modo

Calcula el valor óptimo (límite inferior) para S_k definido como L_k .

3: Si $L_k = I_k$ (*incumbent*) entonces

Borra S_k (No es necesario profundizar)

Se vuelve al paso 1.

De otro modo

Revisar si L_k es una solución factible para el problema original S .

Si es factible entonces

Se define $I_k = L_k$ (nueva solución titular o incumbent)

De otro modo

Se bifurca el problema S_k en S_{k1} y S_{k2} agregándose a la lista de sub-problemas.

4. Se vuelve al paso 1.

El procedimiento BC fue aplicado para el problema del TSP por George Dantzig, Ray Fulkerson y Selmer Johnson en 1964. El concepto aplicado por ellos fue explotar la estructura matemática del problema mediante un procedimiento recursivo a partir de la relajación del problema original y luego resuelto por aproximaciones sucesivas mediante programación lineal y un generador de cortes aplicados de manera incremental en cada iteración. Así el problema original del TSP definido por:

$$\text{Minimizar } c^T x \text{ sujeto a } x \in S.$$

Podría entonces en su versión relajada mediante programación lineal re-expresarse mediante:

$$\text{Minimizar } c^T x \text{ sujeto a } Ax \leq b.$$

La clave del planteamiento anterior enfrentado por Dantzig, Fulkerson y Johnson fue simplemente descubrir cual es el conjunto de restricciones o cortes a incluir en el sistema $Ax \leq b$, que logran que las variables “ x ” satisfagan el requerimiento de integralidad impuesto por “ S ” en el problema original. Efectivamente, la gran contribución de Dantzig, Fulkerson y Johnson fue identificar que al resolver la versión relajada del problema mediante un procedimiento de programación lineal basado en aproximaciones sucesivas, no solamente servía esto para ir progresivamente calculando el límite inferior del problema original. A la vez contribuyó a identificar el sistema de restricciones o cortes que lograban delimitar aquellas soluciones temporales de x^* que no cubrían los requerimientos impuestos por S . Así pues, los espacios solución pertenecientes a soluciones inválidas x^* podrían ser aisladas del espacio S a través de la inclusión progresiva de desigualdades dentro del sistema $Ax \leq b$. A estas desigualdades o cortes se les denomina hiperplanos. La generación de estos hiperplanos eventualmente logran delimitar y ajustar la relajación del problema original.

Así pues, el reto consiste en identificar aquel sistema de desigualdades lineales que satisfagan el conjunto de soluciones válidas incluidas dentro del espacio S pero que a la vez violen la solución óptima x^* . Lo anterior logra que las restricciones de integralidad de las variables discretas del problema original sean resueltas mediante la identificación de los hiperplanos lineales que logren “simular” el espacio solución del problema original. El reto anterior normalmente inicia con la aplicación del sentido común al requerirse explotar las propiedades combinatorias del problema y reflejarlas en su espacio solución S a través del uso de desigualdades lineales ¹³⁴.

¹³⁴ Applegate, David. (1995), Finding Cuts in the TSP, Princeton University, EUA, pag 5.

Lo que se busca es sustituir el sentido común antes mencionado por algún algoritmo que logre generar de manera automática las restricciones o cortes que se requieren para conformar el sistema de desigualdades requerido. Esta necesidad anterior es lo que da origen y sustento al campo de estudio en el área de las matemáticas discretas o combinatorias. Aunque el tratamiento de este último tema esta fuera del alcance del problema de investigación, no obstante se puede revisar la obra de Ralph Gomory quién intentó dar respuesta a esta pregunta a través de sus algoritmos para generación automática de cortes desarrollados en la década de los sesentas ¹³⁵.

La investigación bibliográfica realizada no deja duda acerca de la estrecha relación que existe en las estrategias de corte de Gomory como parte de la estructura algorítmica del método BC. Lo anterior resulta particularmente relevante sobre todo cuando se trata de dar solución a problemas de optimización combinatorios de gran escala ¹³⁶.

Sobre la base de lo antes expuesto, resulta necesario para el investigador, el apoyarse con alguna herramienta (o algoritmo) que le facilite la tarea de generar los hiperplanos que hagan falta durante el proceso de búsqueda de la cota inferior para cada instancia en particular. Así, tal y como se expuso en el apartado 8.1, será un esquema de BC implementado en el código computacional de XPRESS-MP© Versión (13.26), el cual será utilizado por el postulante para dicho propósito. Es precisamente a través de la utilización de dicha infraestructura (software) computacional, mediante la cual podemos obtener provecho del trabajo de investigación de Ralph Gomory ya antes expuesto en el párrafo anterior.

¹³⁵ Gomory, R. (1963), An algorithm for integer solutions to linear programs, Recent Advances in Mathematical Programming (R. L. Graves and P. Wolfe, eds.), McGraw-Hill, New York, pag 269-302.

¹³⁶ Padberg, M; Rinaldi, G. (1991), A Branch and cut algorithm for the resolution of large-scale symmetric traveling salesman problem, "SIAM Review No. 33", EUA, pag 60-95.

8.3 Técnica estadística para la prueba de la hipótesis de investigación.

La técnica que será aplicada para la prueba de la hipótesis central del proyecto será a través de un “Diseño de Experimentos”. La teoría relacionada al tema de “Diseño de Experimentos” es amplia y en términos de su teoría podemos citar al matemático inglés Ronald Aylmer Fisher como uno de sus principales precursores en el siglo XX. Los modelos de “Diseño de experimentos” son modelos estadísticos clásicos cuyo objetivo es averiguar si determinado(s) factor(es) influyen en la variable de interés. El objetivo del diseño de experimentos es determinar si acaso al utilizar un determinado tratamiento, se produce alguna mejora o no en el proceso¹³⁷. Para ello se debe experimentar utilizando un tratamiento y luego otro.

Es comprensible que si se repite un experimento, en condiciones de igualdad donde la mayoría de las causas de variabilidad están controladas (por ejemplo en el laboratorio), los resultados previsiblemente resultarán con una variabilidad pequeña¹³⁸. Uno de los principales objetivos del diseño de experimentos, es precisamente controlar la variabilidad de un proceso estocástico. En general, un experimento está sometido a tres tipos de variabilidad que a continuación se mencionan:

1. Variabilidad sistemática y planificada: esta variabilidad viene originada por la dispersión de los resultados debido a diferencias sistemáticas entre las distintas condiciones experimentales impuestas en el diseño deliberadamente por el investigador. Este es precisamente el tipo de variabilidad que se intenta identificar con el diseño estadístico. Cuando este tipo de variabilidad está presente y es cuantificada, entonces es posible lograr obtener inferencias acerca del proceso en cuestión.
2. Variabilidad no planificada del experimento: esta es la variabilidad no planificada debida a causas desconocidas (ruido aleatorio). Es una variabilidad impredecible. Una manera simple de explicar esta variabilidad es identificándola como la causante de que en un laboratorio, al tomarse medidas repetidas de un mismo experimento, ocurra que en algunos casos, la segunda medida no sea igual a la primera y, más aún, existe la posibilidad del error en la predicción del valor de la tercera. La presencia de esta variabilidad supone la principal causa de conclusiones erróneas. Sin embargo, bajo el aparente caos, existe un patrón regular de comportamiento en esas medidas, todas ellas tenderán a fluctuar en torno a un valor central y siguiendo un modelo

¹³⁷ Fisher, R. (1971), *The Design of Experiments*, Hafner Press & Macmillan Publishers, London, pag 25-50.

¹³⁸ Lorenzen, T. (1993), *Design of Experiments*, Marcel Dekker Inc, EJA, pag 67.

de probabilidad¹³⁹. Si el experimento es bien planificado, entonces será posible estimar dicha variabilidad, lo que permitirá obtener conclusiones y poder hacer inferencias.

3. La tercera fuente de variabilidad proviene del error humano al desarrollar el experimento de una manera desigual al momento de aplicar las mediciones en cada tratamiento.

El diseño de cualquier experimento se desarrolla con la finalidad de obtener mayor conocimiento acerca del comportamiento del proceso de interés. La experimentación forma parte natural de las investigaciones científicas, en muchas de las cuales, los resultados del proceso de interés se ven afectados por la presencia de distintos factores, cuya influencia puede estar oculta por la variabilidad de los resultados muestrales. Así entonces, es fundamental conocer los factores que influyen realmente y así entonces poder estimar esta influencia.

Para conseguir lo anterior, es necesario experimentar. Experimentar significa hacer variar planeadamente las condiciones que afectan a las unidades experimentales y así observar la respuesta en la variable de interés. Del análisis y estudio de la información recogida se obtienen las conclusiones. La forma tradicional que se utiliza en la experimentación, se basa en estudiar las variables controladas (factores) uno a uno, esto significa, hacer variar los niveles de un factor permaneciendo fijos los demás. En cambio, el diseño de experimentos busca sistemáticamente estudiar el efecto de todos los factores.

Aunque la metodología del diseño de experimentos se basa en la experimentación, para que la metodología sea eficaz es necesario que el experimento esté bien diseñado. Así entonces, la metodología del “Diseño de Experimentos” se concentra en establecer las condiciones de realización de un proceso para aumentar la probabilidad de detectar cambios significativos en la respuesta de la variable de interés¹⁴⁰. Dicha metodología se expondrá en el apartado 8.5. Pero antes que exponamos la metodología, definiremos en el siguiente apartado 8.4, la terminología que estaremos empleando a fin de empezar a relacionar el significado de dichos términos respecto a los elementos que estaremos aplicando para el caso de nuestro experimento.

¹³⁹ Hinkelmann, K. (1994), *Design and Analysis of Experiments*, McMillan, EUA, pag 65-70.

¹⁴⁰ Hicks, C. (1994), *Fundamental Concepts in the Design of Experiments*, Holt and Winston Ed, New York, pag 70-96.

8.4 Terminología empleada en el diseño de experimentos.

A continuación enumeramos la terminología que estaremos empleando para explicar el diseño experimental que aplicaremos para nuestro problema de investigación:

1. Unidad experimental: son los objetos, o instancias numéricas en nuestro caso, sobre los que se van a aplicar los experimentos.
2. Variable de interés o respuesta: es la variable que se desea estudiar. En nuestro caso se referirá al % de optimalidad alcanzado.
3. Factor: son las variables independientes (o variables controladas) que se presume pueden influir en la variabilidad de la variable de interés. Los factores son los elementos sobre los cuales el investigador puede hacer las manipulaciones en el experimento.
4. Factor tratamiento: es un factor, del cual el investigador, se interesa en conocer su influencia en la respuesta de la variable de interés. En nuestro caso, existen dos factores para los cuales tenemos el interés en el experimento. El primero se refiere al tipo de algoritmo aplicado y el segundo es el tiempo computacional ejercido en cada tratamiento.
5. Factor bloque: es un factor en el que no se está interesado en conocer su influencia en la respuesta pero se supone que ésta existe y se quiere controlar para disminuir la variabilidad residual. En nuestro caso hablaríamos acerca de los parámetros técnicos de los algoritmos genéticos que estarían siendo fijados como constantes durante todo el experimento.
6. Niveles: estos representan los valores posibles que pueden existir para cada uno de los factores involucrados. Estos niveles, como más adelante se explicará, pueden ser seleccionados por el investigador o elegidos aleatoriamente según sea conveniente en cada caso.
7. Tratamiento: es una combinación específica en la cual se tiene valorizado el nivel correspondiente a cada uno de los factores que están siendo estudiados en el experimento. En un diseño con un factor único, estos son los distintos niveles del factor. Por otro lado, en un diseño con varios factores son las distintas combinaciones de niveles de cada uno de los factores entre sí. Por lo

tanto, un tratamiento es en particular una condición experimental que se desea comparar. En nuestro caso, un tratamiento sería la aplicación de una versión particular de los algoritmos genéticos que vayan a ser confrontados (1er factor), sobre la base de un límite de tiempo computacional también previamente especificado (2do factor).

8. Observación experimental: es cada medición obtenida como respuesta de la variable de interés.
9. Tamaño del Experimento: es el número total de observaciones recogidas en el diseño.
10. Interacción de factores: existe interacción entre dos factores *F1* y *F2*, si el efecto de algún nivel de *F1* se modifica al cambiar el nivel en *F2*.
11. Ortogonalidad de factores: dos factores *F1* y *F2* con *I* y *J* niveles respectivamente, son ortogonales si en cada nivel *I* de *F1* el número de observaciones de los *J* niveles de *F2* están en las mismas proporciones¹⁴¹. En nuestro caso los dos factores, tipo de algoritmo y tiempo computacional, son ortogonales, ya que como se revisará más adelante, existe el mismo número de niveles para cada uno de los factores, es decir, 4 tipos de algoritmos para *F1* y 4 intervalos de tiempo computacional para el factor *F2*.
12. Diseño equilibrado o balanceado: es aquél diseño en el que todos los tratamientos son asignados a un número igual de unidades experimentales. En nuestro caso, tomando en cuenta que tenemos presentes dos factores ortogonales con 4 niveles cada uno, entonces hablamos de que existirán un total de 16 tratamientos a ser aplicados en el experimento. Luego entonces, a cada uno de los 16 tratamientos le aplicaremos un total de 40 instancias de prueba. Por tanto, podemos decir que nuestro experimento posee un diseño equilibrado.

¹⁴¹ Toutenggburg, H. (1995), *Experimental Design and Model Choice*, Physica-Verlag, EUA, pag 125-143.

8.5 Metodología empleada para la planificación del experimento.

A continuación se enumeran las etapas que fueron aplicadas para la planificación de nuestro experimento. Dicha metodología es extraída y resumida de la obra de Weber¹⁴².

1. Definir los objetivos del experimento: debemos definir cuáles son concretamente las preguntas a las que debe dar respuesta en el experimento. Tratar de abordar problemas colaterales puede complicar innecesariamente el experimento. Desde este paso, resulta útil esquematizar el tipo de conclusiones que se espera obtener para poder ir anticipando el modelo o matriz del diseño experimental. Los objetivos más comunes por los cuales se busca hacer un experimento son:
 - a. Determinar las principales causas de variación en la respuesta de la variable de interés.
 - b. Encontrar las condiciones experimentales con las que se consigue un valor extremo en la variable de interés.
 - c. Establecer un modelo estadístico que permita hacer predicciones de respuestas futuras sobre la base de la variable de interés a partir de los parámetros manipulados en el experimento.
 - d. Comparar las respuestas obtenidas en la variable de interés al hacer variar los parámetros (variables controladas) aplicados en cada tratamiento. Este último caso es lo que buscamos cubrir en nuestro proyecto de investigación. En nuestro caso, lo que buscamos determinar es la relación que existe entre el porcentaje de optimalidad alcanzado en los experimentos al aplicar diferentes tipos de instrumentos (algoritmos a ser confrontados) así como también al hacer variar la cantidad de tiempo computacional permitido en cada instrumento. Dicho de otro modo, buscamos conocer la influencia del algoritmo utilizado así como del tiempo de ejecución computacional respecto a la variable de interés (porcentaje de optimalidad).

2. Identificar todas las posibles fuentes de variación:
 - a. Variables controladas (o factores) de tratamiento: un factor es aquella variable fuente cuyo efecto sobre la variable respuesta es de particular interés para el investigador. En nuestro caso existen 2 factores: el tipo de algoritmo experimentado y el tiempo computacional aplicado en cada experimento.
 - b. Niveles de los factores: son los tipos o grados específicos del factor que se tendrán en cuenta en la realización del experimento. Los niveles de estos factores, son básicamente los

¹⁴² Weber, D. (2000), A First Course in the Design of Experiments, CRC Press, EUA, pag 20-78.

grados o valores para los cuales se planea experimentar su efecto en la variable de interés. En nuestro caso, podemos hablar de 4 niveles para el factor del algoritmo computacional y también de 4 niveles para el factor del tiempo computacional. Nótese que, en el diseño de experimentos, es común que los factores cuantitativos sean tratados como cualitativos. Por ejemplo en nuestro caso, los 4 niveles aplicados para el factor del tiempo computacional están codificados de acuerdo a los 4 intervalos de medición previamente diseñados por el investigador. Por lo general, un factor no suele tener más de cuatro niveles¹⁴³.

- c. Factores de ruido: son aquellas fuentes de variabilidad que no son del interés directo para el investigador, pero que se contemplan en el diseño experimental a fin de reducir la variabilidad no planificada. Lo anterior aplicado a nuestro experimento, podemos mencionar que los parámetros técnicos de los algoritmos genéticos, no resultan ser del interés directo para ser experimentados en nuestro proyecto de investigación. Esto significa en nuestro caso que por ejemplo, el tamaño de la población, el porcentaje de cruzamiento, el de mutación y el de elitismo entre otros, son fijados a un mismo valor en todos los algoritmos que van a ser confrontados con la finalidad de eliminar cualquier fuente de variabilidad proveniente de estos factores que no son del interés en este caso.
- d. Características descriptivas de las unidades experimentales: este aspecto aplicado a nuestro caso de investigación, se revisará más adelante en el apartado 8.10.

3. Definir una regla para asignar las unidades experimentales a los tratamientos: la regla de asignación específica cuáles unidades experimentales se observarán bajo cada tratamiento. Las unidades experimentales son el “material” que sirve para evaluar la respuesta de la variable de interés y al que se le aplican los distintos niveles de los factores que estén siendo experimentados. Resulta crucial que las unidades experimentales sean representativas de la población sobre la que se han fijado los objetivos del estudio. Ahora bien, para el caso de nuestro experimento decíamos en el punto anterior que estamos trabajando con 2 factores. Así entonces, un “tratamiento” corresponde a cada una de las combinaciones de los “niveles” que estarán siendo combinados para cada uno de los 2 factores. Luego entonces una “observación” corresponde al resultado obtenido en la variable de interés de acuerdo a una condición específica y determinada por cada uno de los tratamientos que estén siendo aplicados. Las técnicas de asignación se revisarán a detalle en el apartado 8.6.

¹⁴³ Hunter, W. (1986), *Statistics for Experimenters*, John Wiley & Sons, EUA, pag 82.

4. Definir la variable de interés (ó respuesta): es importante precisar de antemano cuál es la variable respuesta y en qué unidades se medirá. Naturalmente, la respuesta está condicionada por los objetivos del experimento. Así por ejemplo en nuestro caso, la variable respuesta estará determinada en una métrica convenientemente establecida en minutos de tiempo computacional.

5. Diseñar el modelo o matriz del experimento: la matriz experimental que se defina, debe indicar la relación que se supone existe entre la respuesta de la variable de interés y las fuentes de variación (factores). En dicha matriz, la respuesta viene dada por una combinación de los factores que representan las principales fuentes de variación planificada. Los modelos de diseño de experimentos se pueden clasificar básicamente en dos tipos:
 - a. Modelo de efectos fijos: es aquel en el cual los niveles de los factores se encuentran predeterminados deliberadamente por el investigador. Este caso resulta apropiado cuando el interés se centra en comparar el efecto de la variable de interés, únicamente en niveles específicos en los factores. Este es precisamente nuestro caso de investigación.
 - b. Modelo de efectos aleatorios: este caso es aquél en el cual para un factor en particular, sólo se incluyen en el experimento una muestra aleatoria simple de todos los posibles niveles que puedan existir para dicho factor. Evidentemente se utilizan estos modelos cuando se tiene un número muy grande de niveles y no es razonable o posible trabajar con todos ellos.

6. Análisis estadístico de la información: en general el análisis de la información dependerá de:
 - a. Los objetivos del experimento determinados en el paso 1. Es necesario delimitar los objetivos del experimento antes de comenzar con el análisis estadístico. Si no se hace así, tratar que el experimento responda a otras cuestiones a posteriori puede ser imposible e incorrecto.
 - b. Del diseño de asignación en las unidades experimentales en los tratamientos visto en el paso 3.
 - c. Del modelo experimental seleccionado en el paso 5.

7. Determinar el tamaño muestral de las unidades a ser aplicadas en el experimento: esto consiste en determinar el número mínimo de observaciones que se deberán aplicar para alcanzar los objetivos del experimento. Existen varios métodos para hacer lo anterior, sin embargo lo que es común en todos estos métodos, es que se requiere el conocimiento del tamaño de la variabilidad no planificada (error experimental). Estimar lo anterior a priori no es fácil. Habitualmente se estima dicho elemento a partir de experimentos pilotos y/o en base a las experiencias previas en trabajos de investigación semejantes.

8.6 Estrategia para la asignación de las unidades experimentales en el experimento.

Para asignar las unidades experimentales a los tratamientos existen tres estrategias básicas las cuales son:

1. Aleatorización: consiste en aleatorizar en las unidades experimentales, todos los factores no controlados por el investigador que pueden influir en los resultados. En este caso, el investigador asigna las unidades experimentales a los tratamientos al azar. La única consideración es el número de observaciones que deberán ser tomadas en cuenta para cada tratamiento.
2. Técnica de bloques: consiste en dividir o particionar las unidades experimentales en grupos llamados bloques, de modo que las observaciones realizadas en cada bloque se realicen bajo condiciones experimentales similares. En este diseño, el investigador agrupa las unidades experimentales en bloques, a continuación determina la distribución de los tratamientos en cada bloque y, por último, asigna al azar las unidades experimentales a los tratamientos dentro de cada bloque. A esta técnica, también se le conoce como diseño “fila-columna”¹⁴⁴, y se caracteriza porque existen unidades experimentales en todas las celdas (intersecciones de fila y columna) en la matriz experimental
3. Diseño factorial: se trata de una estrategia experimental que consiste en cruzar los niveles de todos los factores en todas las combinaciones posibles. Esta técnica permite detectar la existencia de interacción entre los diferentes factores. El diseño factorial resulta muy común encontrarlo en el estudio sobre mejora de procesos industriales (control de calidad). En estos casos es usual trabajar en problemas en los que hay muchos factores que pueden influir en la variable de interés. Los diseños factoriales 2^k son diseños en los que se trabaja con k factores, todos ellos con únicamente dos niveles (binarios). El número de observaciones que necesita un diseño factorial 2^k es muy grande ($n = 2^k$). Precisamente, debido al motivo anterior, la utilización de éste tipo de técnica tiene el gran inconveniente de necesitar un número elevado de observaciones.

Aunque en nuestro experimento no estamos interesados en determinar la existencia de interacción entre los diferentes factores, emplearemos ésta última técnica en virtud de que las 40 unidades experimentales (instancias aleatorias) serán aplicadas a cada uno de los 16 tratamientos que resultan al combinar exhaustivamente los dos factores que estamos considerando en nuestro caso.

¹⁴⁴ Montgomery, D. (1991), Diseño y análisis de experimentos, Grupo Editorial Ibero-Américo, México, pag 82-125.

8.7 Posibles errores en el diseño de experimentos.

La utilización de los modelos de diseño de experimentos se basa en la experimentación y en el análisis de los resultados que se obtienen. Por tal motivo resulta crucial la calidad de los datos. Lo anterior resulta relevante, sobre todo cuando los datos no provienen de un experimento controlado (del laboratorio), sino son extraídos de una fuente de opinión social¹⁴⁵. De lo anterior, resulta importante entonces tomar en cuenta que en el análisis estadístico de los datos pueden aparecer diferentes errores. Los más comunes son:

1. Inconsistencia de los datos: las condiciones en que se aplica cada tratamiento no son equiparables, lo cual ejerce una influencia no deseada sobre los datos recogidos en el experimento. Si derivamos este concepto para nuestro caso de investigación, sería como si a cada algoritmo que va a ser experimentado, lo ejecutáramos a través del uso de procesadores computacionales de velocidad (megahertz) diferentes.
2. Presencia de variables compuestas: puede ocurrir que al experimentar con un proceso existan dos o más variables las cuales en el diseño son establecidas como una sola. Lo anterior puede llevar a situaciones confusas. Por ejemplo, en un proceso existen dos variables $X1$ y $X2$ las cuales influyen en diferente grado en la respuesta de la variable de interés Y , pero si en el diseño del experimento se define solo una variable $X3$ que “intenta” agrupar el efecto de las otras dos, entonces no es posible distinguir si la influencia es debida a una u otra o a ambas variables.
3. Falta de detalle en los datos: se presenta si solo se dispone de datos de una variable (por ejemplo de $X1$ y no de $X2$), lo que puede llevar a pensar que la variable influyente es la $X1$ cuando, en realidad, la variable influyente es la $X2$ (variable oculta).
4. El rango de experimentación de los parámetros es limitado: si el rango de una de las variables controladas en el proceso es insuficiente, entonces no es posible saber su influencia fuera de ese rango y puede quedar oculta, al menos parcialmente, su relación con la variable de interés.

¹⁴⁵ Cochran, W. (1977), *Experimental Design*, John Wiley & Sons, New York, pag 56.

8.8 Definición de los parámetros a ser manipulados en el experimento.

Los parámetros de un experimento, resultan ser los elementos que están bajo el control del investigador. Naturalmente todos estos parámetros están incluidos como parte componente de cada una de las instancias que serán generadas aleatoriamente. En términos generales un parámetro a ser manipulado, necesariamente corresponde a cada una de las variables exógenas del modelo matemático ya presentado en el apartado 6.3. A continuación se enumeran los parámetros que serán manipulados en el experimento. La siguiente lista no pretende volver a enunciar los elementos ya cubiertos en la formulación matemática, pero si al menos complementar la definición de aquellos parámetros que contribuyan a precisar su significado.

1. Cliente: es toda aquella persona que requiere de un servicio de entrega o de recolección de un producto en particular.
2. Sitio o Localización: es todo aquel punto en el cual un cliente requiere que le sea entregado o recogido el producto.
3. Ventana de horario: son los intervalos de tiempo fijados por el cliente durante los cuales es permisible la atención de la entrega o la recolección del producto. Se define entonces que las ventanas de horario se cumplen si y solo si se llega después de la hora de apertura del cliente y antes de la hora de cierre.
4. Demanda: es la cantidad de producto que el cliente requiere le sea entregado o también la cantidad de producto que el cliente requiere le sea recogido.
5. Programa de horarios (“Time Schedule”, en inglés): es la lista de horarios a los cuales se tiene determinado que el vehículo debe llegar en cada parada.
6. Costo de traslado: es el costo, distancia o métrica de esfuerzo incurrido para ir de un cliente a otro dentro de la red.

8.9 Medición de las variables para la prueba experimental.

Las variables son los elementos para los cuales el investigador no tiene acceso a manipular su contenido durante el experimento sino tan sólo a estudiar su comportamiento. Las dos variables principales que se buscan medir en el experimento son:

1. El desempeño o velocidad computacional con la cual el Algoritmo Genético converge en una solución factible. La variable velocidad aparecerá factorizada en el matriz experimental la cual se expondrá en el apartado 8.13.
2. La calidad de la solución respecto a la solución registrada por el grupo control mediante el algoritmo de Ramificación y Corte basado en el BC. A esta medida de calidad le denominamos “porcentaje de optimalidad relativa”. El porcentaje de optimalidad relativa será la variable en donde se aplique la prueba estadística para la hipótesis (revisar apartado 8.13).

Es importante aquí precisar, que el presente experimento, no tiene el objetivo de incluir a “*w*” (*cantidad de nodos en cada instancia*) como parte de las variables a ser medidas en el experimento. Dicho de otro modo, no se busca establecer en el experimento, el como el valor de “*w*” (*donde: 100 = w = 120*), influye en la variable “velocidad” y/o en la variable “calidad” para las instancias que sean aplicadas.

Así, mientras en el Algoritmo Genético se busca medir con qué rapidez se mejora una solución, con el Algoritmo de Ramificación y Corte basado en el BC se obtendrá el valor de referencia (o grupo control) para poder medir el desempeño del Algoritmo Genético propuesto. Lo anterior ratifica la importancia que la aplicación del Algoritmo de Ramificación y Corte basado en el BC tiene para nuestro experimento, ya que de otra manera no se tendría disponible una referencia cuantitativa para la prueba de la hipótesis de investigación.

De lo anterior, resulta interesante determinar la relación que guarda el tiempo de ejecución computacional versus el porcentaje de optimalidad alcanzado. Lo anterior servirá para establecer la relación costo-beneficio entre el tiempo que se tiene disponible para ejercer la toma de decisiones versus el nivel de calidad deseado. La comprensión de ésta relación resulta ser crítico cuando se trata de implementar algoritmos que tendrán que ofrecer soluciones en ambientes en tiempo real.

8.10 Estudio descriptivo de las instancias numéricas a ser utilizadas para la prueba experimental.

Aunque se expondrá este tema en detalle más adelante en el apartado 8.12, podemos en este momento establecer que será necesario generar instancias numéricas para cubrir el requerimiento de las unidades experimentales para poder desarrollar el experimento. Las instancias numéricas (o datasets), tendrán que ser aleatoriamente generadas y serán utilizadas como unidades experimentales para ser aplicadas tanto en el Algoritmo de Ramificación y Corte (BC) que servirá para la integración del grupo control, así como también en el método propuesto basado en un Algoritmo Genético.

Para lo anterior será necesario desarrollar un generador computacional que permita explotar la dimensionalidad cuantitativa de las instancias numéricas que serán utilizadas para el experimento. En la conformación de las instancias a ser obtenidas a través del generador aleatorio, es necesario que se consideren algunos aspectos que le darán validez al experimento. A continuación se enumeran los principales:

1. El cálculo aleatorio de las variables exógenas del problema, se asume que deban ser números positivos y enteros.
2. Se requiere que la desigualdad del triángulo se cumpla tanto para la matriz de costos C_{ij} , así como también para la matriz de tiempos T_{ij} , esto para cada uno de los nodos de la red. Esto matemáticamente se expresa como:

$$\{ : i,j,k \in V \ C_{ij} + C_{jk} = C_{ik} \text{ y también } T_{ij} + T_{jk} = T_{ik} \}$$

3. Lo antes expuesto para la conformación de la instancia, no debe ser visto como una limitante en el alcance práctico del planteamiento del problema. Después de todo resulta natural ya que para trasladarse de un nodo “ i ” a nodo “ k ” siempre resultará más económico hacerlo de manera directa que a través de la intermediación de un 3er. nodo “ j ” que solo sirva como medio de interconexión. De hecho esta propiedad aquí definida, será de utilidad ya que nos permitirá incluir en la variable T_{ij} , no solamente el tiempo que se ocupa para trasladarse del nodo “ i ” al nodo “ j ”, sino también eventualmente el tiempo que se ocupa en el cliente “ i ” en ser atendido ya sea por (1) el proceso de entrega o, (2) el proceso de recolección.

4. Se requiere que las instancias generadas cubran una fase de pre-procesamiento aplicado con la finalidad de explotar la estructura del problema. El principal objetivo de la fase de pre-procesamiento es construir una formulación equivalente al problema original pero que logre conformar su espacio solución lo más ajustado posible.
5. Cuando nos referimos a tener un espacio solución ajustado, nos referimos a identificar aquellos cortes (o hiperplanos) que a su vez ofrezcan una cota inferior y superior al problema que logren acelerar el proceso de búsqueda de la solución óptima en el espacio solución. El reto de lo anterior consiste en que:
 - a. La formulación ajustada del problema debe asegurar no perder el acceso a la solución óptima del problema original a la vez que,
 - b. La solución óptima del problema ajustado debe mantenerse siendo también una solución óptima para el problema original.

En cuanto a la fase de pre-procesamiento a ser incluida en el generador de instancias aleatorias, podemos a su vez enumerar las siguientes estrategias:

1. Procedimiento para reducción de amplitud en las ventanas de horario (“Time Windows Reduction”, en inglés): este procedimiento tiene la finalidad de acotar o cerrar lo más posible la amplitud de las ventanas de horario definidas para cada cliente. Lo anterior se desarrolla de acuerdo a las restricciones de precedencia que resulten por consecuencia lógica de la revisión de las propias ventanas de horario. Este procedimiento resulta crucial ya que está probado científicamente que existe una dificultad intrínseca en relación al tamaño de la ventana de horario. Entre más amplia sea la ventana de horario, más grande es el espacio solución a tener que ser explorado y por tanto más difícil resulta el problema en ser resuelto¹⁴⁶.
2. Procedimiento para construcción de red (“Network Construction”, en inglés): busca obtener ventaja de los resultados obtenidos en el procedimiento anterior de reducción de las ventanas de horario. El objetivo es conformar una red que incluya solo aquellos arcos que resulten factibles de acuerdo a las ventanas de horario definidas en cada uno de los nodos, así como también de acuerdo a las relaciones de precedencia y a las restricciones de capacidad. La explotación de la estructura lógica anterior, tiene el objetivo de lograr formular una red mucho más esbelta, ya que logra identificar una gran cantidad de arcos que pueden ser omitidos del planteamiento del modelo matemático lo cual redundará en aspectos de eficiencia computacional.

¹⁴⁶ Desrosiers, J; Dumas, Y; Solomon, M; Soumis, F. (1995), Time constrained routing and scheduling, G.L. Nemhauser editor, Handbooks in Operations Research and Management Science, Holanda, pag 23-30.

8.11 Cálculo para el tamaño de la muestra.

El cálculo del tamaño de muestra “*n*” equivale en nuestro caso a la determinación de la cantidad de instancias aleatorias que serán utilizadas para el experimento. Si ya antes habíamos precisado que el rigor del experimento dependía de la calidad de los integrantes en el grupo control, entonces tenemos que precisar que la validez del experimento dependerá de que la cantidad de instancias que sean empleadas para el experimento, sea lo suficientemente representativo. En nuestro caso podemos afirmar que la cantidad de elementos en la población es infinito “?”. Es comprensible entonces pensar que va a ser necesario determinar un tamaño de muestra suficiente para lograr alcanzar la validez del experimento.

Para el cálculo del tamaño de muestra existen diversos procedimientos. Uno que es muy simple de utilizar es a través de la aplicación de la siguiente formula:

$$n = \frac{z^2 \cdot p \cdot q}{B^2}$$

De la formula anterior, a continuación explicamos en significado y el valor de cada variable:

z = Nivel de Confiabilidad = este valor esta asociado al área bajo la curva de distribución normal que se tiene a partir de cierta cantidad de desviaciones estándar “?”. Así por ejemplo si se quiere establecer en el experimento una confiabilidad del **95%**, entonces quiere decir que solo nos vamos a permitir un 5% de probabilidad por arriba o abajo (prueba de dos colas) del criterio marcado. **Así pues, un 95% de confiabilidad equivale a 1.96 desviaciones estándar.**

p = Probabilidad de éxito = este valor representa la probabilidad esperada de que un evento puntual del experimento resulte favorable. Este valor proviene de la empírica del investigador así como de las evidencias de éxito en experimentos provenientes de otras investigaciones. Así entonces, si especificamos un **95%**, quiere decir entonces que vamos a comprometer este valor para su verificación al final del experimento.

q = Probabilidad de fracaso = este valor representa la probabilidad estimada por el investigador de que un evento puntual en el experimento resulte fuera de lo deseado.

$B^2 = \text{Error experimental} =$ es el valor del error experimental que nos vamos a permitir.
 Un error experimental menor al 5%, es una excelente medida.

Si entonces aplicamos los parámetros anteriores a la formula obtenemos entonces:

$$n \approx \frac{z^* p^* q}{B^2} \approx \frac{1.96 * (0.95) * (0.05)}{(0.05)^2} \approx \frac{0.0931}{0.0025} \approx 37.24 \approx 38$$

Como ya antes mencionábamos, existen otros métodos para calcular el tamaño de la muestra. Por ejemplo, si nosotros ya conociéramos, en base a un pre-muestreo, información adicional acerca de cómo se comportan ambas poblaciones (la del grupo control basada en el BC y la del grupo perteneciente al Algoritmo Genético), pues entonces podríamos aplicar una metodología diferente. A continuación en la Figura 8.2, se muestra el cálculo para el tamaño de la muestra basado en el conocimiento de la información de ambas muestras.

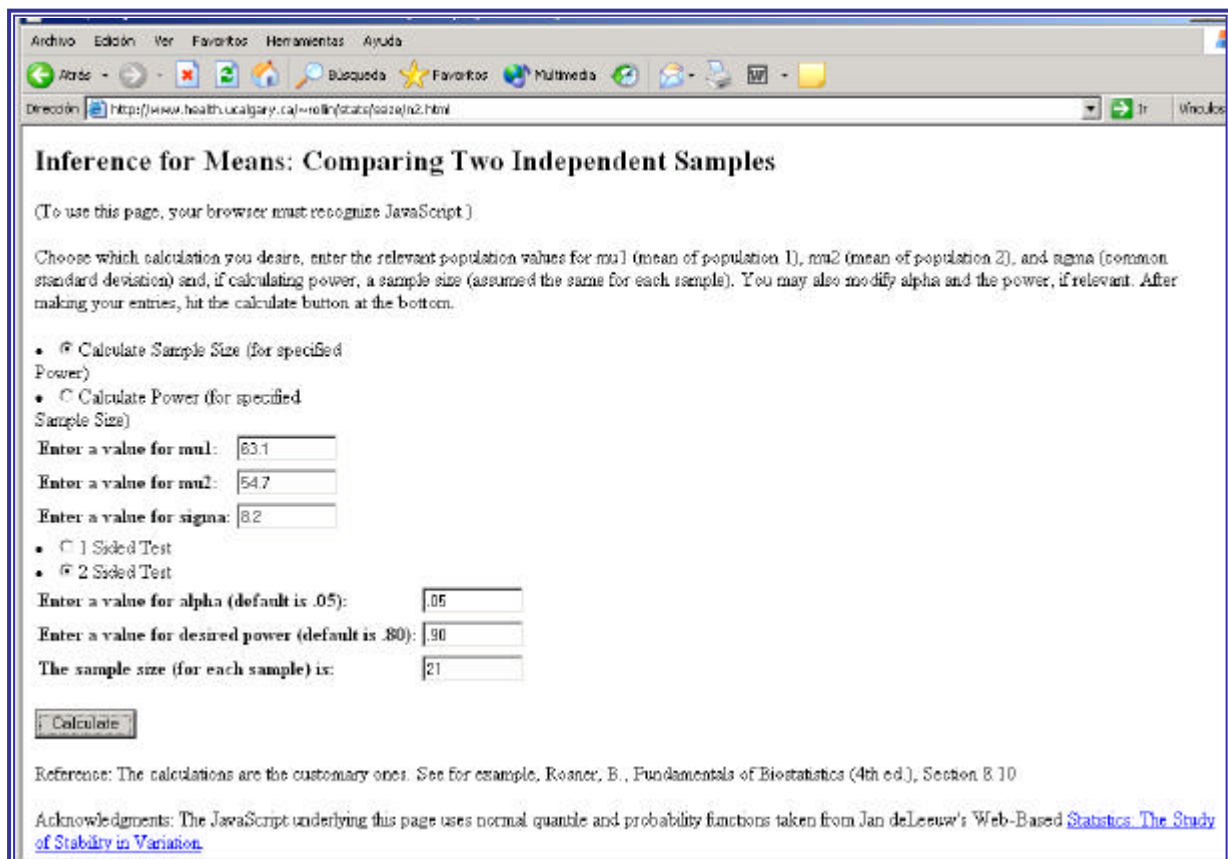


Figura 8.2 Cálculo del tamaño de la muestra a partir de información disponible de las muestras ¹⁴⁷.

¹⁴⁷ Disponible en <http://www.health.ucalgary.ca/>

Como puede verificarse en la figura anterior, es fácil darse cuenta que el tener información a partir de un pre-muestreo resulta útil, ya que logra que el tamaño de la muestra pueda disminuirse en favor del costo y del tiempo requerido para la ejecución del experimento. No obstante se puede verificar que el tamaño de la muestra, *al disminuir de 38 a solo 21 unidades experimentales*, también pierde la propiedad de normalidad que se obtiene al manejar muestras por arriba de 30 elementos.

El postulante establece manejar el experimento a través de un tamaño de muestra suficientemente grande como para cubrir el requerimiento de ambos procedimientos. Más aún, *se emplearán un total de 40 unidades experimentales*, con la tranquilidad y soporte que nos brinda el haber revisado los dos procedimientos alternos. Recordemos que la dimensionalidad (cantidad de nodos a ser tratados) para cada una de las 40 instancias, ya fue definido en el apartado 7.4 ($100 = w = 120$).

8.12 Planteamiento de la prueba experimental.

Definiremos para propósitos de nuestro diseño experimental, que el desempeño de un algoritmo está determinado por la efectividad del mismo. Ahora bien, cuando nos referimos en nuestro experimento a la efectividad del algoritmo, estamos hablando de las siguientes características a ser cubiertas:

1. Eficiencia en el tiempo de ejecución computacional: = *5 minutos*
2. Calidad en las soluciones arrojadas: *porcentaje de optimalidad > 90%*
3. Cantidad de nodos a ser considerados en la instancia: *100 = w = 120*

Así, el mecanismo mediante el cual podemos probar la efectividad de un algoritmo propuesto para solucionar un problema determinado, pues es precisamente el diseño de la prueba experimental. El diseño de la prueba experimental estará determinado por el desempeño que muestre la aplicación del Algoritmo Genético a través de la manipulación de los parámetros del proceso de optimización. En nuestro caso, el diseño experimental está constituido por dos pruebas a ser aplicadas a cada instancia:

1. Prueba de Desempeño Computacional: consiste en medir el valor de la función objetivo que resulta después de una cantidad de tiempo fija y predeterminada de esfuerzo computacional.
2. Prueba de Calidad para la Exploración del Espacio Solución: consiste en medir que tan cerca se termina respecto a la solución óptima para un problema determinado después de una cantidad de tiempo fija de esfuerzo computacional. La medición de esta prueba se lleva a cabo mediante el cálculo de un valor denominado “Porcentaje de Optimalidad Relativa”, el cual se calcula de la siguiente manera:

$$\text{porcentaje de optimalidad} = 1 - \frac{(\text{solución alcanzada} - \text{cota inferior})}{\text{cota inferior}} \times 100\%$$

relativa

Por ejemplo, supongamos que para una instancia de prueba, la cota inferior obtenida mediante la aplicación del algoritmo BC resultase 529.12 (costo de la ruta). Luego supongamos que la solución alcanzada mediante la aplicación del algoritmo genético propuesto fuese 544.89, por tanto aplicando la fórmula del “porcentaje de optimalidad relativa” obtendríamos un valor de 97.02%. Es comprensible darse cuenta que, para el valor del “porcentaje de optimalidad relativa”, entre más se acerque al 100%, será indicativo de una mayor cercanía entre la solución alcanzada y la cota inferior (solución óptima). También

es fácil darse cuenta que este “porcentaje de optimalidad relativa”, nunca podrá ser mayor al 100%, ya que esto significaría que el algoritmo genético propuesto estaría encontrando un valor “mejor” que el referenciado en la cota inferior, lo cual equivale a invalidar al grupo control.

Ambas pruebas experimentales contribuyen a lograr los siguientes objetivos:

1. Calibrar los parámetros del Algoritmo Genético: es decir, establecer aquellos parámetros, o combinación de ellos, que hagan que el algoritmo alcance su máximo desempeño.
2. Contrastar la solución obtenida mediante el Algoritmo Genético con respecto a la solución del grupo control obtenida a través del algoritmo basado en el BC.

La metodología general para la medición de las variables a ser monitoreadas durante la prueba experimental es como sigue:

1. Se generan “n” instancias Aleatorias con valores uniformemente distribuidos. Estas instancias vendrán a conformar el conjunto de las unidades experimentales. El valor de “n” es precisamente el “Tamaño de la Muestra” experimental ya antes tratado.
2. Para cada una de las “n” instancias obtenidas en el punto 1, se ejecuta computacionalmente el algoritmo basado en el BC. Para cada ejecución computacional se aplicarán 4 mediciones en intervalos de tiempo predeterminados los cuales se definirán más adelante.
3. Para cada una de las 4 mediciones a realizar en el punto 2, se mide el valor de la función objetivo que haya sido alcanzada hasta este momento y ésta se registra como la mejor solución encontrada hasta cada momento correspondiente.
4. Si después de haber transcurrido la 4ta medición aún no finaliza el algoritmo BC su proceso de exploración para la solución óptima, entonces se deja continuar el algoritmo hasta su finalización con un límite de tiempo a 5 horas. Una vez que el algoritmo BC concluye, el valor reportado en la función objetivo, se establece como la cota inferior aplicada para cada unidad experimental que esté siendo tratada.
5. Las cotas inferiores registradas en el punto 4 integrarán el grupo control del experimento.

6. Para cada una de las “n” instancias generadas en el punto 1, de manera equivalente se ejecuta computacionalmente el Algoritmo Genético. Al igual que en el tratamiento del grupo control, para cada ejecución computacional se aplicarán 4 mediciones en los mismos intervalos de tiempo predeterminados.

7. El resultado de cada una de las corridas desarrolladas en el punto 6 vendrá a conformar el grupo de los tratamientos experimentales a ser comparadas con respecto al grupo control obtenido en el punto 5.

8.13 Implementación del experimento.

Conviene recordar que en el aparatado anterior se definieron dos pruebas para el diseño del experimento. Los resultados a ser obtenidos en la 1era prueba experimental, referente al desempeño computacional del algoritmo, serán tratados a través de una exposición descriptiva que permita visualizar el desempeño obtenido en cada una de las 4 mediciones a ser aplicadas para cada uno de los 4 algoritmos y a su vez para cada una de las unidades experimentales que vayan a ser probadas.

De este modo entonces, será la 2da. prueba experimental que ya antes denominamos como “Prueba de Calidad para la Exploración del Espacio Solución” la que será aplicada de manera específica para la prueba de la hipótesis de investigación. Recordaremos a su vez que la variable que en esta prueba vamos a medir es la que nombramos como “porcentaje de optimalidad relativa”. La medición de esta variable se hará mediante la aplicación de los 4 instrumentos experimentales. A continuación se enumera cada uno de estos:

1. Algoritmo de Ramificación y Corte (BC) como método de solución exacta. Este algoritmo es que el contribuirá a la conformación del grupo control que será tomado en cuenta en relación a los otros tres algoritmos.
2. Algoritmo Genético básico implementado mediante el motor (o software) de optimización “Evolver” © Versión 6.0 de Palisade Corporation.
3. Algoritmo Evolutivo básico implementado mediante el motor (o software) de optimización “Solver” © Versión 4.0 de Frontline System.
4. Algoritmo Evolutivo en versión avanzada (propuesta) e implementado mediante el motor (o software) de optimización “Solver” © Versión 4.0 de Frontline System.

El 1er. algoritmo (BC) correspondiente al grupo control, tendrá el propósito de establecer la referencia de optimalidad para cada una de las unidades experimentales (o instancias). El algoritmo BC se aplicará directamente sobre la base del problema original, es decir, en la estructura de la red sin compactar, a diferencia de cómo ocurrirá con el algoritmo propuesto #4. Los instrumentos #2, #3 y #4 serán entonces aplicados para verificar su desempeño en la búsqueda de la solución óptima que haya sido registrada mediante el algoritmo #1. El 4to. instrumento, corresponde precisamente a la propuesta algorítmica del proyecto de investigación. Esta última será entonces comparada contra las otras dos implementaciones básicas, la primera del algoritmo genético (2) y la segunda del algoritmo evolutivo (3).

En lo referente al algoritmo genético propuesto #4, su tiempo de ejecución computacional corresponde al tiempo total incurrido en las 6 fases expuestas ya anteriormente en el apartado 7.7 de la propuesta metodológica. Debemos recordar que en las 6 fases algorítmicas, tenemos 4 fases de pre-procesamiento una más de post-procesamiento, aparte por supuesto de la fase evolutiva.

Para cada uno de los 4 instrumentos antes descritos, la medición del “porcentaje de optimalidad relativa” será aplicado en 4 momentos sucesivos de tiempo durante la ejecución de cada experimento:

- a. Después de transcurrir el minuto #3 de ejecución computacional.
- b. Al minuto # 5.
- c. Al minuto # 8.
- d. Al minuto # 10.

El 1er. instrumento basado en el algoritmo BC, no está sujeto a un límite de tiempo, ya que su objetivo es obtener la solución óptima para cada unidad experimental. De lo anterior debemos suponer que las instancias que serán aplicadas deberán tener una complejidad matemática lo suficientemente intensa como para requerir el consumo de un mínimo de 10 minutos de esfuerzo computacional para alcanzar la solución óptima mediante el algoritmo BC.

A continuación se resumen las características dimensionales del diseño del experimento:

1. Cantidad de instrumentos a ser aplicados = 4 instrumentos / experimento
2. Cantidad de mediciones que serán aplicadas = 4 mediciones / instrumento
3. Cantidad de tiempo requerido en cada experimento = al menos 10 minutos / medición
4. Cantidad de unidades experimentales = 40 instancias

De lo anterior, tenemos entonces un experimento de 6400 minutos computacionales ($4 \times 4 \times 10 \times 40$). Lo anterior equivale a 107 horas (o casi 5 días naturales) de recurso computacional.

Concluidas las mediciones a ser aplicadas durante la prueba de calidad para la exploración del espacio solución ya antes descrito, se procede a calcular para cada unidad experimental el “porcentaje de optimalidad relativa”. Así, para cada una de las unidades experimentales (instancias) se tendrán entonces 16 valores cuantitativos correspondientes al cálculo de “porcentaje de optimalidad relativa”. La razón por la cual son 16, es debido a que estamos hablando de 4 instrumentos de medición (4 algoritmos) y además para cada uno de estos se aplicarán 4 mediciones en los intervalos de tiempo que ya antes explicamos. Sí

estamos hablando de 16 mediciones para cada una de las 40 unidades experimentales, entonces estamos hablando de un total de 640 mediciones, es decir 160 mediciones por cada instrumento algorítmico.

Así pues, para cada una de las 640 mediciones, la variable de “porcentaje de optimalidad relativa” será calculada a través del procedimiento que ya antes describimos. El siguiente paso del diseño del experimento consiste en estratificar las 640 mediciones a través de 4 grupos de 160 mediciones cada uno. Cada agrupación corresponde a las mediciones correspondientes a cada uno de los 4 instrumentos algorítmicos que están siendo comparados en el experimento. Luego a su vez, las 160 mediciones de cada uno de los 4 instrumentos, requiere volver a estratificarse en 4 grupos de 40 mediciones. En ésta ocasión, cada uno de los 4 grupos corresponde a las 4 mediciones que estamos aplicando para cada instrumento y a su vez a cada unidad experimental.

Así obtenemos 16 (4 x 4) grupos de 40 mediciones (unidades experimentales). Dadas las condiciones de normalidad que fueron procuradas al establecer el tamaño de la muestra en el apartado 8.11, el siguiente paso consiste en que para cada uno de los 16 grupos de 40 mediciones, se procede a calcular los siguientes parámetros estadísticos aplicados a la variable “porcentaje de optimalidad relativa”:

1. Media muestral (m).
2. Desviación estándar muestral (s).

La matriz del diseño del experimento, consiste en una tabla de 4 filas por 4 columnas. Cada una de las 4 filas de la matriz representa a cada una de las 4 mediciones que son hechas en cada unidad experimental. Por otro lado tenemos que cada una de las 4 columnas de la matriz representa a cada una de los 4 instrumentos que están siendo comparados. En la Tabla 8.1, se muestra lo anterior.

Tabla 8.1 Matriz resultante del diseño del experimento.

		Instrumentos de Medición a ser comparados			
		Algoritmo BB (Grupo Control)	Algoritmo Genético Básico (Evolver)	Algoritmo Evolutivo Básico (Frontline)	Algoritmo Evolutivo Avanzado (Frontline)
Mediciones a ser aplicadas después de cierto tiempo computacional	3er Minuto	(m_{11} , s_{11})	(m_{12} , s_{12})	(m_{13} , s_{13})	(m_{14} , s_{14})
	5to Minuto	(m_{21} , s_{21})	(m_{22} , s_{22})	(m_{23} , s_{23})	(m_{24} , s_{24})
	8avo Minuto	(m_{31} , s_{31})	(m_{32} , s_{32})	(m_{33} , s_{33})	(m_{34} , s_{34})
	10mo Minuto	(m_{41} , s_{41})	(m_{42} , s_{42})	(m_{43} , s_{43})	(m_{44} , s_{44})

A su vez, el significado de cada una de las celdas de la matriz corresponde al cálculo de la Media muestral (m_{ij}) así como a la Desviación estándar muestral (s_{ij}). Así por ejemplo (m_{ij}) sería la media muestral verificada para el instrumento algorítmico “j” en la medición del intervalo de tiempo computacional “i”.

La prueba estadística que utilizaremos para la comprobación de la hipótesis será a través de la aplicación de una Prueba “T”. Dicha Prueba “T” se aplicará de manera individual a cada uno de los 16 elementos de la matriz. La aplicación de la Prueba “T” en cada estadístico (m_{ij}, s_{ij}) calculará la probabilidad de que las mediciones obtenidas mediante la aplicación del instrumento algorítmico “j” y en el intervalo de tiempo computacional “i” hayan obtenido un “porcentaje de optimalidad relativa” mayor al 90%. Dicho de otro modo, buscamos calcular para cada uno de los 16 elementos de la matriz el valor de $P(x > 90\%)$, donde “P(x)” es el área bajo la curva para el estadístico de una distribución “T” de Student. De esta manera ahora el diseño de nuestra matriz se presenta a continuación:

Tabla 8.2 Matriz de probabilidades para la prueba de la hipótesis.

Prueba T de Student para la Variable "% de Optimalidad Relativa"					
		Instrumentos de Medición a ser comparados			
		Algoritmo BB (Grupo Control)	Algoritmo Genético Básico (Evolver)	Algoritmo Evolutivo Básico (Frontline)	Algoritmo Evolutivo Avanzado (Frontline)
Mediciones a ser aplicadas después de cierto tiempo computacional	3er Minuto	$P(X11 > 90\%)$	$P(X12 > 90\%)$	$P(X13 > 90\%)$	$P(X14 > 90\%)$
	5to Minuto	$P(X21 > 90\%)$	$P(X22 > 90\%)$	$P(X23 > 90\%)$	$P(X24 > 90\%)$
	8avo Minuto	$P(X31 > 90\%)$	$P(X32 > 90\%)$	$P(X33 > 90\%)$	$P(X34 > 90\%)$
	10mo Minuto	$P(X41 > 90\%)$	$P(X42 > 90\%)$	$P(X43 > 90\%)$	$P(X44 > 90\%)$

Finalmente entonces, la prueba de la hipótesis de investigación puede ser resuelta verificando los porcentajes de probabilidad que se obtengan para cada uno de los elementos en la matriz. Así por ejemplo, si $P(X34 > 90\%) = 93\%$, nosotros entonces podríamos inferir con un 93% de certeza que el Algoritmo Evolutivo Avanzado (4to instrumento) puede mejorar el 90% de optimalidad relativa en un tiempo menor a los 8 minutos de ejecución computacional (3era medición).

Tomando en cuenta el ejemplo anterior así como la estructura de la Tabla 8.2, se puede comprobar que en ningún momento se está haciendo referencia a la dimensionalidad ($100 = w = 120$) de las instancias que serán aplicadas en el experimento. Lo anterior es debido a que este atributo no será considerado como una variable a ser medida en el experimento. No obstante, es oportuno mencionar que para las 40 instancias a ser aplicadas en el experimento, el grado de dimensionalidad de cada instancia estará determinado a partir de una variable aleatoria “ w ” uniformemente distribuida entre 100 y 120. Tomando en cuenta lo anterior es predecible que para cada valor de “ w ” entre 100 y 120, existan dos instancias generadas que logren de este modo cubrir las 40 instancias (tamaño de la muestra “ n ”) para el experimento.

En el siguiente capítulo 9 nos abocaremos a exponer los resultados del experimento aquí descrito.

9. Resultados.

9.1 Introducción.

Empezaremos por definir las condiciones computacionales que fueron aplicadas durante el experimento:

1. Sistema operativo “Windows XP ©”
2. Procesador computacional INTEL © Pentium 4 a 2.4 Ghz (Gigahertz).
3. Memoria de acceso inmediato de 128 Mhz (Megahertz).

Los 4 instrumentos de medición fueron implementados computacionalmente. Las instancias aplicadas en el experimento fueron desarrolladas a partir de redes arriba de 100 nodos ($100 = w = 120$). A este respecto es importante recordar que en el trabajo de Ascheuer, Fischetti, y Grotschel en el 2001, ellos exponen en base a sus experiencias computacionales que las instancias del TSP-TW en el límite de 50 hasta 70 nodos, pueden ser resueltas hasta la solución óptima por el algoritmo de Ramificación y Corte BC. De esta manera, nuestro algoritmo para el Grupo Control basado también en el BC, sobrepasó esta frontera antes especificada.

Por igual, recordemos que en los resultados obtenidos en el trabajo de Ascheuer, Fischetti, y Grotschel en el 2001 se concluye que, en general, las instancias para el problema de investigación resultan particularmente difíciles de resolver para aquellas en las que la cantidad de nodos activos que contengan restricciones de ventana de horario estén por arriba del 50%. A ese respecto, nuestro experimento aplicó el tratamiento a unidades experimentales todas por arriba del 70% con nodos activos. Nuevamente, el reto computacional para nuestro algoritmo de Ramificación y Corte BC consistió en ir más allá de este límite antes expuesto.

Finalmente y no menos importante es el tema de la amplitud de las ventanas de horario para los nodos activos. Recordemos que matemáticamente hablando, entre más cerradas sean las ventanas de horario más pequeño es el espacio solución a ser explorado y por tanto más fácil es el problema en resolver. En el experimento desarrollado, todas las unidades experimentales que fueron utilizadas, además de tener

al menos un 70% de nodos con ventanas de horario activas, también éstas tienen al menos un 75% de amplitud en sus ventanas. Esto quiere decir que las instancias estuvieron caracterizadas por ventanas de horario con un alto porcentaje de traslape lo cual hace más difícil el problema para ser resuelto en términos de su solución óptima.

De todo lo anterior expuesto podemos confirmar que la condición de complejidad matemática estuvo plenamente cubierta y por tanto la prueba aplicada a nuestro Algoritmo Genético avanzado (4to instrumento de medición) fue significativa. Los resultados obtenidos en el experimento fueron satisfactorios.

Previo al análisis estadístico de la hipótesis mediante la aplicación de la prueba “T” de Student, estableceremos la siguiente Tabla 9.1 para dar a conocer los resultados generales del experimento:

Tabla 9.1 Matriz de frecuencias para los 16 grupos experimentales.

Resultados: Matriz de Frecuencias $F(x > Opt\%)$

		Instrumentos de Medición a ser comparados					
		Algoritmo BB (Grupo Control)	Algoritmo Genético Básico (Evolver)	Algoritmo Evolutivo Básico (Frontline)	Algoritmo Evolutivo Avanzado (Frontline)		
					F(x > 90%)	F(x > 92.5%)	F(x > 95%)
Mediciones a ser aplicadas después de cierto tiempo computacional	3er Minuto	NA	15	13	40	33	18
	5to Minuto	2	19	16	40	38	22
	8avo Minuto	6	23	25	40	40	33
	10mo Minuto	12	26	32	40	40	38

9.2 Prueba de la hipótesis de investigación.

Finalmente se procede a aplicar la prueba de la hipótesis. Los métodos estadísticos disponibles para hacer análisis básicamente son de dos tipos:

1. Los métodos descriptivos.
2. Los métodos inferenciales

A su vez los métodos estadísticos inferenciales pueden ser aplicados para dos propósitos muy bien diferenciados:

1. Pruebas de Hipótesis.
2. Estimación de intervalos de confianza

En nuestro caso lo que aplica es la prueba de hipótesis y para ello existen diversos métodos estadísticos que pueden ser de utilidad. Básicamente existen aquellos que tienen que ver con (1) Estadística Paramétrica y (2) Estadística NO Paramétrica. La estadística paramétrica es más potente debido a los supuestos de los cuales se parte al momento de iniciar el análisis de la muestra. Dichos supuestos son tales como el tamaño de la muestra o también alternativamente la propiedad de normalidad intrínseca de la población. En nuestro caso y tomando en cuenta ambos aspectos recién mencionados, haremos uso de la Prueba estadística “T” de student.

La prueba estadística “T” de student, a diferencia de la prueba “Z”, es más rigurosa debido a que no presupone el conocimiento acerca del valor de la desviación estándar de la población. En sustitución utiliza la desviación estándar de la muestra. No obstante, lo anterior tiene un costo estadístico. Lo anterior se manifiesta a través de los “Grados de Libertad” (df en inglés) que el estadístico va a permitir en los datos de la muestra. La prueba “T” que va a ser aplicada para la prueba de nuestra hipótesis será a través de una prueba de diferencias de medias. Los supuestos de los cuales se parte para dicho análisis son los siguientes:

1. Las poblaciones de donde parten las instancias que serán aplicadas para los 4 instrumentos algorítmicos están normalmente distribuidas.
2. Las muestras u observaciones que serán medidas están apareadas.
3. La desviación estándar de las poblaciones es desconocida.
4. Las muestras que serán tomadas para cada tratamiento es de tamaño 40.

Esencialmente en toda prueba de hipótesis es necesario comparar el valor del estadístico calculado versus el valor crítico del estadístico. En nuestro caso, la fórmula para calcular el valor del estadístico “T” para una prueba de diferencia de medias es como sigue a continuación:

$$t = \frac{\bar{D} - \mu_D}{\frac{S_D}{\sqrt{n}}}, \text{ donde: } \bar{D} = \frac{\sum_{i=1}^n D_i}{n}, \text{ y } S_D = \sqrt{\frac{\sum_{i=1}^n (D_i - \bar{D})^2}{n - 1}}$$

A continuación en la Tabla 9.2 (a), se muestran los valores calculados del estadístico “T”.

Tabla 9.2 (a) Matriz de valores calculados para el estadístico “T”.

		Instrumentos de Medición a ser comparados					
		Algoritmo BB (Grupo Control)	Algoritmo Genético Básico (Evolver)	Algoritmo Evolutivo Básico (Frontline)	Algoritmo Evolutivo Avanzado (Frontline)		
					P(x > 90%)	P(x > 92.5%)	P(x > 95%)
Mediciones a ser aplicadas después de cierto tiempo computacional	3er Minuto	NA	-0.4039	-0.5581	2.4261	1.0688	-0.0911
	5to Minuto	-2.4260	-0.1159	-0.3068	3.3130	1.5392	0.1111
	8avo Minuto	-1.2800	0.1616	0.3173	4.8510	4.1050	0.8300
	10mo Minuto	-0.7001	0.4001	0.9030	6.2980	5.5770	1.3277

Los valores calculados de “T” son los que se comparan contra los valores críticos de “T” a efectos de determinar la aceptación o el rechazo de la hipótesis nula. Por regla general tenemos que:

Si $T_{Calc} > T_{Crítico}$ entonces Se acepta H_0
en caso contrario
Se acepta H_a

Habitualmente el valor de $T_{Crítico}$ se puede obtener a través del uso de tablas estadísticas o alternativamente mediante software computacional. Invariablemente, se requiere considerar dos elementos. El primer elemento es el nivel del error permitido α (alfa) y el segundo elemento son los grado de libertad de la prueba (df). En cuanto al valor de α (alfa), este típicamente se considera sobre

la base de un 5% a un 1% de error. En otras palabras, si α (alfa) es el nivel del error permitido, entonces $1 - \alpha$ equivale al nivel de confiabilidad buscado en la prueba, o sea de un 95% a un 99% de confiabilidad. Por otro lado, la definición de los grados de libertad (*df*) se define de acuerdo al tipo de prueba que se va a implementar (cantidad de muestras involucradas), así como también, de acuerdo al tamaño de la muestra(s). Recordemos que en nuestro caso particular, la prueba a implementar es:

$$H_0: \mu_{\text{Grupo experimental}} = 90\% (\mu_{\text{Grupo Control}})$$

$$H_a: \mu_{\text{Grupo experimental}} \neq 90\% (\mu_{\text{Grupo Control}})$$

Como se puede apreciar en la formulación anterior, se trata de una prueba estadística de una cola y con una sola muestra a comparar con respecto al grupo control. Por tal motivo podemos entonces definir que los grados de libertad en nuestro caso se define como: $df = n - 1$, donde $n =$ tamaño de la muestra. El uso de los grados de libertad (*df*), es lo que hace diferenciar la prueba “T” de la prueba “Z”. La distribución estadística en una prueba “Z” está representada por una distribución normal única. En cambio en la prueba “T”, la distribución estadística no es única ya que se trata en realidad de una familia de distribuciones cuasi-normales, en donde, cada una de estas distribuciones está diferenciada por los grados de libertad (*df*) que definen la prueba a realizar.

A medida que los grados de libertad de la muestra disminuyen, entonces las áreas bajo la curva de distribución que están contenidas en las colas aumentan. En términos generales, podemos decir que los valores críticos de la prueba “T” van aumentando en similitud a los valores críticos de la prueba “Z” en la medida en la cual el tamaño de la muestra crece por arriba de los “30” elementos. Para ejemplificar la similitud anterior, haremos uso de nuestra prueba de hipótesis una sola “cola”, en la cual las muestras experimentales son de tamaño “40”. Esto quiere decir que el grado de libertad (*df*) equivale a “39” en nuestro caso. A partir de lo anterior tendríamos los siguientes valores críticos:

	<u>$\alpha = 5\%$</u>	<u>$\alpha = 1\%$</u>
Valores Críticos de Z:	1.645	2.33
Valores Críticos de T:	1.685	2.426

Con lo anterior podemos concluir que: a medida que disminuyen los elementos de la muestra, menor son los grados de libertad de la prueba y por ende, crecen los valores críticos de la prueba “T” que son requeridos para lograr el rechazo de la hipótesis nula.

A partir de los resultados expuestos en la Tabla 9.2 (a), en donde se expresaron los valores calculados del estadístico “T”, es posible entonces para cada elemento de la matriz, ir aceptando o rechazando la hipótesis de optimalidad según corresponda en cada caso. Tan solo hace falta comparar dichos valores calculados que aparecen en la tabla 9.2 (a) versus el valor crítico de “T” que nosotros aspiremos a probar. Si buscáramos un 95% de confiabilidad entonces la referencia mínima sería “1.685”. Si buscáramos un 99% de confiabilidad entonces la referencia mínima sería “2.426”.

En lugar de verificar tan solo la hipótesis a través de la comparación del valor calculado de “T” versus su valor crítico, alternativamente es posible para cada valor calculado de “T” establecer su correspondiente valor de significancia o confiabilidad. Esto quiere decir que aunque en algunos casos no sea posible alcanzar un nivel α que sea suficiente para probar una hipótesis, si al menos es posible establecer su nivel de significancia (*P Valor*). A continuación, en la Tabla 9.2 (b) se muestran los porcentajes de probabilidad “P Valor” resultantes de la prueba “T” de student aplicados en el diseño del experimento para cada uno de los 16 grupos de la matriz. Estos valores de “P” corresponden uno a uno a los valores calculados del estadístico “T” que se reportaron en la Tabla 9.2 (a).

Tabla 9.2 (b) Matriz experimental de probabilidades “P Valor” para el Estadístico “T”, $P(x > \text{Opt}\%)$.

		Instrumentos de Medición a ser comparados					
		Algoritmo BB (Grupo Control)	Algoritmo Genético Básico (Evolver)	Algoritmo Evolutivo Básico (Frontline)	Algoritmo Evolutivo Avanzado (Frontline)		
					P(x > 90%)	P(x > 92.5%)	P(x > 95%)
Mediciones a ser aplicadas después de cierto tiempo computacional	3er Minuto	NA	34%	29%	99%	85%	46%
	5to Minuto	<1%	45%	38%	100%	93%	54%
	8avo Minuto	10%	56%	62%	100%	100%	79%
	10mo Minuto	24%	65%	81%	100%	100%	90%

A partir de los resultados numéricos expuestos en la tabla 9.2 (b), procedemos a continuación a desarrollar una breve explicación de los mismos:

1. El Algoritmo de Ramificación y Corte BC, como era de esperar, invariablemente obtiene sino la solución óptima, si al menos una mejor solución que nuestro Algoritmo Genético. Recordemos que para el caso del Algoritmo de Ramificación y Corte BC, aunque será utilizado como grupo control, no es requerido en el objetivo del proyecto de investigación comprobar su optimalidad.
2. Los tiempos computacionales registrados para el Algoritmo de Ramificación y Corte BC, oscilan entre los 20 minutos y las 5 horas dependiendo de cada unidad experimental que haya sido tratada.
3. Es interesante puntualizar que nuestra implementación algorítmica resulta estar mucho más influenciada por la cantidad de nodos con ventanas de horario activas que por la amplitud de las ventanas de horario. Esta propiedad resulta entonces útil por la característica abordada en el proyecto de investigación al hablar del término “Ventanas de tiempo negociables”.
4. En cuanto al experimento con las implementaciones basadas en Algoritmos Genéticos, debemos iniciar precisando que todos los parámetros (Tamaño de la población, Método de Selección, Operador de Cruce, porcentaje de Mutación y porcentaje de Elitismo) de los Algoritmos Genéticos correspondientes a los instrumentos 2, 3 y el 4, fueron fijados al mismo valor con la finalidad de hacer comparables sus desempeños.
5. Los valores que fueron fijados para los parámetros de los Algoritmos Genéticos provienen de la empírica del investigador así como también de los trabajos de investigación ya revisados en la bibliografía. El resumen de los valores aplicados a dichos parámetros puede ser revisado en la figura 7.1, la cual se expone en el apartado 7.7 correspondiente a la propuesta metodológica.
6. La implementación de nuestro Algoritmo Genético avanzado (instrumento # 4), logra consistentemente obtener soluciones satisfactorias (*arriba del 90% de optimalidad*) y en tiempos de ejecución computacional razonables (= *3 t = 5 minutos*). Es importante agregar que estadísticamente hablando, las mediciones anteriores tienden a mejorar al hablar de sus promedios por razones derivadas del teorema del límite central.
7. Para el caso de los Algoritmos Genéticos Básicos (instrumento # 2 y # 3), los tiempos de convergencia registrados (tiempo computacional después del cual ya no hay mejora en la solución) son más o menos parecidos a los observados en el instrumento # 4. No obstante los porcentajes de optimalidad medidos resultan ser significativamente inferiores ya que estos nunca sobrepasan del 90%. Más bien tienden a oscilar entre el 80% y el 88% de optimalidad.

8. Es importante destacar en términos generales que aunque el instrumento # 2 (Algoritmo Genético de reproducción de estado estable) “evoluciona” mucho más rápido que el instrumento # 3 (Algoritmo Genético de reproducción generacional), el instrumento # 3 ofrece al final mejores porcentajes de optimalidad que el # 2.
9. De lo anterior podemos precisar que, aunque en los primeros 3 minutos de ejecución computacional el instrumento # 2 ofrece mejor desempeño que el instrumento # 3, en el experimento se observa que en el primer caso el algoritmo se “atasca” en un óptimo matemático local resultado de una degeneración prematura en la población de organismos.
10. La situación anterior es aprovechada por el instrumento # 3 el cual aunque empieza en las primeras generaciones muy lento, continúa mejorando la población de organismos y aprovechando el tiempo perdido por el instrumento # 2 durante la fase terminal de degeneración en el óptimo local.
11. Tomando en cuenta que deliberadamente todos los parámetros ya antes expuestos de los 3 Algoritmos Genéticos fueron fijados a un mismo valor, entonces la diferencia observada en el desempeño de estos 3 instrumentos puede ser inducida de la siguiente manera:
 - a. El instrumento # 3 obtiene mejores soluciones que el instrumento # 2 debido a la propiedad generacional en su método de reproducción.
 - b. El instrumento # 4 (propuesto) obtiene mejores soluciones que los otros dos debido a que aprovecha del instrumento # 3 la propiedad generacional de reproducción ya antes expuesta y además debido a la explotación en la estructura del problema que resulta ser el fundamento de la investigación doctoral en la que se aplica un algoritmo genético para resolver un problema muy particular.
12. Se aplicaron de manera complementaria algunas instancias “sencillas” para observar el desempeño del Algoritmo de Ramificación y Corte BC. Las instancias sencillas a las que nos estamos refiriendo podemos describirlas de manera general como aquellas por debajo de los 70 nodos y con menos del 60% de nodos activos en sus ventanas de horario. El resultado verificado en estos experimentos es un desempeño extraordinariamente rápido y eficiente del algoritmo. Para este tipo de instancias, se ocuparon tiempos computacionales por debajo de los 3 minutos.

9.3 Análisis costo-beneficio para la aplicación del Algoritmo Genético propuesto.

Las variables empleadas en el diseño experimental antes expuesto, tiempo computacional y % de optimalidad, han sido ampliamente promovidas para su uso en proyectos de investigación previos¹⁴⁸. Dicho diseño experimental, tuvo el objetivo de comprobar el desempeño que nuestra propuesta algorítmica ofrece con respecto al grupo control determinado por un método de solución exacta.

Las justificaciones del proyecto de investigación, expuestas en el capítulo 7, apuntan claramente a identificar la necesidad de herramientas de optimización para apoyar a la función de la logística de ruteo. No obstante lo anterior, es claramente verificable que el objetivo del proyecto de investigación nunca fue establecer una métrica para analizar el costo-beneficio que nuestra propuesta ofrece como ventaja económica al ser implementada en una problemática real de negocio .

El tratamiento del análisis costo-beneficio de la propuesta sobre la base de una justificación económica, puede ser resuelta a través de comparar el beneficio obtenido, versus el costo económico que tuviera que ser erogado para obtener dicho beneficio. Así entonces, en un sentido práctico diríamos que la propuesta es económicamente viable, si el beneficio económico obtenido por la propuesta algorítmica es “razonablemente” mayor que el costo requerido para su implementación.

El investigador desarrollará a continuación un análisis costo-beneficio, sobre la base de una serie de supuestos cuantitativos que han sido definidos, deliberadamente en una forma conservadora, con la finalidad de lograr minimizar el error que pueda existir en las conclusiones que de aquí se generen. Dichos supuestos provienen de la empírica del investigador, los cuales serán expuestos numéricamente a lo largo de la exposición de las Tablas 9.3, 9.4 y 9.5 a continuación.

Comenzamos con la Tabla 9.3 en la cual se exponen los elementos que el investigador está proponiendo para la implementación de la propuesta de solución. En dicha tabla, cada elemento involucrado se acompaña con una estimación típica del costo en cada caso correspondiente. Los costos están estipulados pensando en la implementación de la propuesta en una empresa con operaciones en el área metropolitana de ciudad de Monterrey (AMM).

¹⁴⁸ Cheng, R (1997), Genetic Algorithms and Engineering Design, John Wiley Interscience, Boston Ma, EUA, pag 43-57.

Tabla 9.3 Costos relacionados a la propuesta de solución basado en la aplicación de un Algoritmo Genético para solucionar el problema de ruteo SPDP-sTW en una empresa del AMM.

	COMPONENTES	INVERSION	DEPRECIACION / GASTO ANUAL	Depre- ciación
Costo de Oportunidad: Profesionista competente en la aplicación del OR/MS	Profesionalización de la función de ruteo Logístico	1 Coordinador	\$240,000	
	Prestaciones		\$120,000	
	Habilitadores: Seminarios y Talleres		\$10,000	
	SUB-TOTAL	67%	\$370,000	
Costo Tecnológico: Hardware, Software, Investigación y Desarrollo	Computadora Pentium IV (>=2.0 GHz)	\$25,000	\$6,250	4 Años
	Mapas Digitales (calles de la Cd. Monterrey)	\$20,000	\$10,000	2 Años
	Licenciamiento x uso de Algoritmo Genético	\$200,000	\$66,667	3 Años
	Soporte Técnico y Capacitación		\$100,000	
	SUB-TOTAL	33%	\$182,917	
COSTO TOTAL DEL PROYECTO			\$552,917	

Como se puede apreciar en la tabla anterior, el costo relacionado al desarrollo de un profesionista que logre aplicar la propuesta algorítmica para la obtención de rutas de distribución mejoradas, ocupa el 67% del total del costo del proyecto. El desarrollo de dichas competencias en el profesionista no es trivial, ya que como se verá más adelante en el apartado 9.4, la configuración apropiada de los parámetros en el algoritmo genético será crucial para mejorar la relación costo-beneficio del proyecto. Hablar de que las dos terceras partes (67%) del costo del proyecto, recaen en el ámbito de la aplicación del algoritmo genético por parte de un profesionista con características específicas, justifican en gran medida la importancia de haber incluido el factor educativo y cultural del elemento humano en los antecedentes del proyecto de tesis.

Una vez cubierto el componente del costo en la justificación económica, entonces lo que procede a continuación, es dar tratamiento al componente del beneficio esperado. El cálculo del beneficio esperado puede ser atendido en dos aspectos principales los cuales son:

1. Eficiencia en el uso de las capacidades de transporte.
2. Efectividad en los horarios de servicio comprometidos con el cliente.

a. Eficiencia en el uso de las capacidades de transporte:

La función objetivo del problema SPDP-sTW, está determinado por la minimización del costo de distribución en el ruteo propuesto. De esta manera es fácil entender que, al rutear óptimamente un vehículo de transporte en términos del costo de distribución, entonces puede existir la factibilidad en tiempo y en capacidad de carga, de que una mayor cantidad de clientes puedan ser cubiertos con el mismo recurso de transportación. A su vez, esto quiere decir que al haber más clientes siendo atendidos por cada vehículo, entonces los vehículos se vuelven más productivos. Si los vehículos son más productivos, entonces el administrador de un negocio puede optar por hacer una de dos alternativas:

- a. Aprovechar el exceso de capacidad de transporte para incrementar el nivel de cobertura en la distribución y ventas del negocio.
- b. Optimizar (disminuir) la flota de equipo de transporte.

Ahora bien, los beneficios provenientes de la primera alternativa son difíciles de cuantificar, ya que en la realidad, no existiría evidencia suficiente acerca de cuantos clientes, y por consiguiente qué volumen de venta adicional podría llegar a materializarse sobre la base de solo un potencial excedente en la capacidad de transportación. Dicho de otro modo, sería más conservador e ineludible, disponer de un recurso de transporte solo hasta que los requerimientos del mercado así lo exigieran. Sobre el razonamiento anterior aplicaremos entonces la segunda alternativa, en la cual, los beneficios resultan mucho más objetivos y tangibles. A continuación en la Tabla 9.4, se muestran los costos anuales que típicamente podrían ser evitables por cada ruta que fuese disminuida, como consecuencia de la optimización en la capacidad de transporte vía un ruteo más eficiente.

Tabla 9.4 Ahorro potencial por ruta a ser disminuida de la flota vía eficiencia en el ruteo.

COMPONENTES DEL COSTO DE UNA RUTA	INVERSION	Depreciación	Depreciación / Gasto Anual	% Aplicación	Costo Anual
Vehículo de Transporte (Camión + Carrocería)	\$500,000	10 Años	\$50,000	100%	\$50,000
Mantenimiento			\$20,000	50%	\$10,000
Combustible			\$30,000	20%	\$6,000
Personal Operador	1 Chofer + 1		\$240,000	100%	\$240,000
Prestaciones	Ayudante		\$120,000	100%	\$120,000
TOTAL: GASTO DE OPERACIÓN ANUAL POR RUTA DE TRANSPORTE					\$426,000

b. Efectividad en los horarios de servicio comprometidos con el cliente:

El tratamiento de las ventanas de horario a ser consideradas para el servicio de entrega y recolección a los clientes, no forma parte de la formulación del costo económico en la función objetivo del problema SPDP-sTW. No obstante, esta consideración aparece formulada como una restricción dentro del modelo matemático del problema (revisar apartado 6.2). Lo anterior significa que, cualquier solución de ruteo que proponga el algoritmo genético, cumplirá rigurosamente el aspecto de los horarios de servicio para la totalidad de los “*n*” clientes que estén siendo tratados en la instancia del problema.

De acuerdo a lo anterior y dejando de lado cualquier circunstancia estocástica que pudiera presentarse en la realidad, para las soluciones generadas por el algoritmo genético podemos afirmar que, la probabilidad de llegar fuera del horario estipulado es del 0%. De acuerdo a la caracterización de las instancias que han sido atendidas en el presente proyecto de investigación (revisar apartado 9.1), tenemos que la cantidad promedio de clientes por ruta es de 100 nodos, con al menos un 70% de nodos activos con ventanas de horario. Esto quiere decir que tenemos 70 clientes a los que debemos arribar dentro de un horario de atención específico. Entonces la pregunta que nos tendríamos que hacer sería:

¿Cual es la probabilidad de que un despachador de rutas en una empresa, sin herramientas de optimización, logré secuenciar un vehículo de transporte atinando a los horarios de llegada de 70 clientes que requieren considerar ventanas de horario?

Dar respuesta a lo anterior resulta arriesgado. Para ejemplificar esto, podemos partir de un escenario conservador en el cual, gracias a la experiencia del despachador, éste secuencie las llegadas a los clientes de tal manera que solo en 1 cliente de cada 100, no logre empatar el horario de llegada requerido. Así entonces, si la confiabilidad en el procesamiento de cada ventana de horario correspondiente a cada cliente fuese del 99%, entonces tendríamos que la probabilidad de que una ruta estuviese libre de errores sería igual a: $99\%^{70} \approx 49\%$. De lo anterior se puede concluir que, estimar un beneficio por disponer de una herramienta de optimización equivale a estimar el costo de oportunidad por dejar de vender producto por no llegar a tiempo a la cita. Ambos elementos siguen siendo subjetivos a la vista de los cuestionamientos. Para remedir lo anterior, lo que proponemos hacer es recurrir a los registros históricos de la compañía en búsqueda de las ventas perdidas que pudiéramos definir que existen por motivo del producto que las rutas devuelven.

Vamos a continuación a generar los siguientes supuestos para el análisis:

1. Suponemos que solo el 50% del producto devuelto sea imputable a las llegadas fuera de horario.
2. Si de éste, solo el 50% pudiera ser resuelto a través de la implementación de un ruteo más eficaz, entonces tendríamos un potencial del 25% de reducción en el producto que se devuelve.
3. Si el porcentaje de producto devuelto que una empresa actualmente tiene fuese del 10%, entonces a través de un ruteo más inteligente podríamos aumentar las ventas en un 2.5%.
4. Si la contribución neta a utilidades (EBITDA) por la venta del producto fuese, digamos del 15%, entonces tendríamos un incremento del 0.4% en las utilidades netas de la compañía.
5. Si partimos de una estructura de costos en la cual, el costo de operación de una ruta no debiera exceder del 20% de lo que vende, entonces tendríamos que una ruta, como la que se expuso en la tabla 9.4, debería estar vendiendo aprox. \$2 Millones al año (ó \$6,500 por día).
6. Si una empresa opera con 50 rutas, entonces ésta facturaría \$100 Millones al año.
7. Con lo anterior, tendríamos un beneficio marginal sobre utilidades de \$375,000 por año.

Con la finalidad de resumir los supuestos comentados, en la Tabla 9.5 se exponen los factores que hacen sensibilizar el beneficio potencial a obtenerse mediante el uso del algoritmo propuesto.

Tabla 9.5 Análisis Costo-Beneficio derivado de la implementación del algoritmo propuesto.

Matriz Análisis Costo-Beneficio (Miles \$ / Año)			% de Incremento en Ventas (Mejoramiento en el Servicio)				
			1%	2%	3%	4%	
ECONOMIA DE ESCALA DE LA EMPRESA	50 Rutas	% de Rutas a Rutas a Optimizar	1%	-\$190	-\$40	\$110	\$260
			2%	\$23	\$173	\$323	\$473
			5%	\$662	\$812	\$962	\$1,112
			10%	\$1,727	\$1,877	\$2,027	\$2,177
	100 Rutas	% de Rutas a Rutas a Optimizar	1%	\$173	\$473	\$773	\$1,073
			2%	\$599	\$899	\$1,199	\$1,499
			5%	\$1,877	\$2,177	\$2,477	\$2,777
			10%	\$4,007	\$4,307	\$4,607	\$4,907
	200 Rutas	% de Rutas a Rutas a Optimizar	1%	\$899	\$1,499	\$2,099	\$2,699
			2%	\$1,751	\$2,351	\$2,951	\$3,551
			5%	\$4,307	\$4,907	\$5,507	\$6,107
			10%	\$8,567	\$9,167	\$9,767	\$10,367
	500 Rutas	% de Rutas a Rutas a Optimizar	1%	\$3,077	\$4,577	\$6,077	\$7,577
			2%	\$5,207	\$6,707	\$8,207	\$9,707
			5%	\$11,597	\$13,097	\$14,597	\$16,097
			10%	\$22,247	\$23,747	\$25,247	\$26,747
Costo implementación del algoritmo (\$ x Año) =			\$552,917				
Costo de Operación por Ruta (\$ x Año) =			\$426,000		21%		
Venta por Ruta (\$ x Año) =			\$2,000,000				
Contribución Marginal de Ventas a Utilidad =			15%				

Como se puede apreciar en los escenarios de sensibilidad expuestos en la tabla anterior, existen básicamente 3 factores que están siendo combinados para efectos de estimar el costo-beneficio:

1. Economía de escala de la empresa (# de Rutas en operación)
2. Factor de Rutas a optimizar (optimización de la capacidad de transporte)
3. Factor de Ventas a incrementar (optimización en el servicio al cliente)

En cuanto al 1er factor, no queda duda de que entre mayor sea la escala de operación de la empresa mayor es el beneficio potencial. En cuanto al tratamiento de los otros dos factores, hasta este momento, todo está analizado numéricamente en el ámbito de los supuestos ya antes mencionados, lo cual no deja de ser inquietante, ya que:

“No es posible a priori, saber que tan lejos de la solución óptima puede estar cualquier ruta que actualmente esté operando en las empresas de distribución, sobre la base de una práctica operativa basada tan solo en la experiencia de las personas”.

Estimar la brecha (o el grado de mejoramiento), que pueda existir entre una ruta desarrollada en forma manual versus el óptimo matemático, requeriría la aplicación de una de las siguientes dos estrategias:

1. Desarrollar un estudio comparativo para calcular el efecto (supuesto beneficio) que se obtendría al sustituir las rutas que actualmente estén siendo operadas en una empresa por las soluciones que sean obtenidas mediante la propuesta basada en el algoritmo genético.
2. Desarrollar un estudio analítico que permita estimar el comportamiento que pudiera ser obtenido derivado del uso del algoritmo, a través del estudio de la superficie de respuesta del algoritmo genético.

En lo referente al uso de la primera estrategia, si quisiéramos generar resultados concluyentes respecto a la brecha de optimalidad, tendríamos que desarrollar un muestreo para varias empresas. La razón de lo anterior, vuelve a ser sobre la base de que las rutas que actualmente se estén generando en forma empírica por una persona, seguramente serían diferentes en función a la experiencia de otra persona. Por la razón anterior y de acuerdo al alcance ya antes definido en el objetivo de la tesis, el investigador aplicará la segunda estrategia para lograr generar datos más concluyentes al respecto.

No obstante lo anterior, el investigador ha desarrollado un breve pero significativo análisis comparativo acerca de la brecha de optimalidad que puede existir entre un conjunto de rutas que anteriormente estuvieron en operación con respecto a lo que el algoritmo genético propuso como ruta mejorada.

Dicho análisis se aplicó para una muestra de 20 rutas pertenecientes a una misma compañía de distribución de producto con operaciones en el área metropolitana de la ciudad de Monterrey (AMM). Los resultados comparativos se exponen a continuación en la Tabla 9.6

Tabla 9.6 Análisis comparativo de los beneficios obtenidos en términos de las distancias a recorrer entre una ruta de distribución actual versus la ruta optimizada propuesta por el algoritmo genético.

Ruta muestreada	Kms de Trayecto del almacén a la zona	Kms de Ruteo en la Zona		Beneficio imputable al Ruteo		Kms Totales de Transportación		
		Actual	Optimizado	Beneficio en Kms	Beneficio %	Actual	Optimizado	Beneficio %
1	11.6	6.6	5.0	1.6	24%	18.2	16.6	9%
2	10.9	9.1	6.9	2.2	24%	20.0	17.8	11%
3	10.1	23.1	21.1	2.0	9%	33.2	31.2	6%
4	20.6	12.4	7.9	4.5	36%	33.0	28.5	14%
5	28.8	9.4	8.0	1.4	15%	38.2	36.8	4%
6	28.4	11.4	8.9	2.5	22%	39.8	37.3	6%
7	22.5	7.3	5.4	1.9	26%	29.8	27.9	6%
8	14.8	19.4	16.2	3.2	16%	34.2	31.0	9%
9	7.4	10.2	9.2	1.0	10%	17.6	16.6	6%
10	8.6	9.5	5.8	3.7	39%	18.1	14.4	20%
11	22.0	8.3	6.1	2.2	27%	30.3	28.1	7%
12	19.0	14.3	11.4	2.9	20%	33.3	30.4	9%
13	12.0	12.2	8.3	3.9	32%	24.2	20.3	16%
14	13.6	6.8	4.7	2.1	31%	20.4	18.3	10%
15	20.8	7.1	5.2	1.9	27%	27.9	26.0	7%
16	20.5	6.6	4.3	2.3	35%	27.1	24.8	8%
17	8.7	21.3	14.8	6.5	31%	30.0	23.5	22%
18	8.4	7.3	4.7	2.6	36%	15.7	13.1	17%
19	9.6	4.6	3.7	0.9	20%	14.2	13.3	6%
20	14.3	9.9	7.6	2.3	23%	24.2	21.9	10%
Prom	15.6	10.8	8.3	2.6	25%	26.5	23.9	10%

Como puede apreciarse en la tabla anterior, existen dos perspectivas de beneficio que pueden ser útiles para efectos de cuantificar la cantidad de recursos de transporte que pueden ser optimizados (reducidos) de la flota. El primero está sobre la cantidad de kilómetros que recorren los vehículos durante su ruteo entre cada uno de los clientes que están incluidos dentro de la zona (columnas 5 y 6). La columna 5 hace referencia al beneficio absoluto medido en kilómetros, mientras que la columna 6 lo re-expresa en términos porcentuales. La segunda perspectiva del beneficio potencial esta determinada

en la última columna de la tabla, la cual hace referencia a la cantidad de kilómetros totales que recorre la ruta, incluyendo los trayectos desde que sale del almacén hasta que regresa al mismo. En ese sentido obtenemos un 10% de beneficio global en comparación con el beneficio imputable únicamente al ruteo entre cliente y cliente, el cual alcanza hasta un 25% de beneficio global para la muestra de las 20 rutas.

El análisis comparativo anterior, sea con el beneficio al 25% o al 10%, no deja duda de que conservadoramente podamos alcanzar al menos un 10% de beneficio en el tema de reducción de costos económicos debido a la optimización de la flota de transporte. Dicha argumentación apoya plenamente el análisis costo-beneficio expuesto ya antes en la Tabla 9.5. En dicha tabla puede ser verificado que se desarrollaron 4 escenarios de optimización económica, desde un 1% hasta un 10% de brecha de optimalidad.

Con todo lo expuesto hasta este momento podemos resumir lo siguiente en referencia al análisis costo-beneficio de la propuesta:

1. De acuerdo al análisis expuesto en la tabla 9.5, la conveniencia económica para la aplicación del algoritmo genético está influenciado básicamente por tres factores:
 - a. Economía de escala de la empresa (# de Rutas en operación = EE)
 - b. Factor de Rutas a optimizar (optimización de la capacidad de transporte = FR)
 - c. Factor de Ventas a incrementar (optimización en el servicio al cliente = FV)
2. Aún en empresas debajo de 50 rutas ($EE = 50$), resulta recomendable la aplicación del algoritmo. Su contribución se verifica significativamente para valores de $FR = 5\%$. Alternativamente también puede verificarse dicha contribución para valores de $FV = 3\%$ y de una manera mucho más importante cuando ambos factores se conjugan.
3. El 1er factor referido a la economía de la escala de la empresa (EE), es el que resulta más significativo al hablar de la conveniencia económica de la aplicación del algoritmo genético.
4. Del factor (EE), tenemos que para el 2do cuartil, es decir $50 = EE = 100$, el máximo beneficio económico sería de \$4.9 Millones y el promedio de \$2.1 Millones por año.
5. Para el 3er cuartil ($100 = EE = 200$), tenemos que el máximo beneficio económico sería de \$10.4 Millones y el promedio de \$4.8 Millones por año.
6. Finalmente, para el 4to cuartil ($200 = EE = 500$), tenemos que el máximo beneficio económico sería de \$26.7 Millones y el promedio de \$12.8 Millones por año.

Con la tranquilidad que nos brinda el haber aplicado un muestreo para calcular el beneficio esperado, ahora la investigación se dispone a aplicar la segunda estrategia para lograr estimar datos complementarios que permitan ser concluyente al respecto de la justificación económica de la propuesta algorítmica. No debe quedar duda de que esta metodología es completamente válida. Lo anterior se fundamenta, sobre la base siguiente:

“Cualquier ruta, presuntamente no óptima, que actualmente esté siendo operada en una empresa sin el uso del algoritmo propuesto, está incluida dentro del espacio solución descrito por el poliedro del problema SPDP-sTW. A su vez, la topología de dicho poliedro, puede ser analizada a través del desarrollo de un modelo de regresión multi-variado. Para dicho modelo tenemos que:

- 1. La variable dependiente corresponde a la función objetivo del problema SPDP-sTW.*
- 2. Las variables independientes (parámetros estadísticos) corresponden a los parámetros de operación del algoritmo genético.*

El análisis costo-beneficio expuesto ya anteriormente en la Tabla 9.5 y verificado mediante un muestreo en la Tabla 9.6, debiera ser suficiente para evidenciar una justificación económica para la aplicación del algoritmo genético propuesto, al menos en términos prácticos. No obstante, el investigador está convencido de que la brecha de optimalidad que existe entre una ruta generada mediante métodos empíricos versus aquella obtenida mediante métodos matemáticos, es un valor diferencial que sigue siendo desconocido y que por tanto es interesante generar algún esfuerzo al menos para predecirlo. Para ello el investigador desarrollará un análisis experimental el cual será cubierto en los apartados 9.4 y 9.5. Dicho análisis experimental permitirá estudiar de una manera mucho más científica dicho potencial de mejora.

Así entonces, a continuación en el apartado 9.4, se exponen las bases sobre las cuales haremos la experimentación para el estudio analítico de los parámetros de operación del algoritmo genético. Los parámetros de operación del algoritmo genético a ser utilizados para el desarrollo del modelo de predicción serán expuestos más adelante en el apartado 9.5.

9.4 Estudio analítico para los parámetros de operación del Algoritmo Genético.

El objetivo primordial de cualquier estudio analítico consiste en determinar los parámetros y sus valores que mejor ofrecen un desempeño de operación buscado. Lo anterior resulta contrastante al revisar la falta de investigación enfocada en utilizar dichas métricas para el desarrollo de análisis comparativos entre diferentes tipos de meta-heurísticas para resolver problemas de naturaleza combinatoria¹⁴⁹. De lo anterior, tan solo se disponen de pocas referencias a proyectos tales como el desarrollado en 1995 para resolver el famoso problema combinatorio de Steiner. En dicho proyecto de investigación, Esbensen desarrolla un análisis comparativo entre el desempeño de un algoritmo genético propuesto versus dos heurísticas conocidas y ambas en relación a la solución óptima global obtenida mediante un procedimiento de solución exacta basado en algoritmos de ramificación y corte.

Los resultados de Esbensen apuntan a concluir que las soluciones obtenidas mediante la aplicación de su algoritmo genético, logran consistentemente alcanzar soluciones muy cercanas al óptimo global para instancias de pequeña dimensionalidad. En cuanto a las instancias de media o alta dimensionalidad, Esbensen concluye que el algoritmo genético logra obtener mejores soluciones que las heurísticas comparadas en el análisis comparativo. La razón de lo anterior se concluye, es debido a que el esfuerzo computacional requerido por un algoritmo genético se incrementa a una velocidad menor que respecto el crecimiento del espacio solución del problema¹⁵⁰. Dicha propiedad no ocurre con el procedimiento de solución exacta de ramificación y corte. Este último resulta extremadamente eficiente para problemas de baja dimensionalidad, pero su desempeño se deteriora rápidamente de acuerdo con el crecimiento del espacio solución.

No obstante la virtud anteriormente expuesta para el uso de los algoritmos genéticos en lo relacionado al tema de la dimensionalidad, esto no resulta evidencia suficiente para afirmar que este tipo de algoritmos puedan ser aplicados de manera inmediata a cualquier tipo de problema. Lo antes mencionado no es trivial sobre todo en problemas de investigación como el nuestro, en donde el espacio solución descrito por el poliedro, se encuentre delimitado y sometido a un conjunto importante de restricciones (revisar apartado 6.2). Tan solo hace falta pensar en la cantidad tan enorme de organismos, que debido a las restricciones del poliedro, seguramente resultarán infactibles a lo largo

¹⁴⁹ Reeves, C. (1995), Genetics algorithms flow shop sequencing, Computers & Operations Research, Vol 22, EUA, pag 5-13.

¹⁵⁰ Esbensen, H. (1995), Computing near-optimal solutions to the Steiner problem using genetics algorithms, Networks Interscience, Vol 26, No.4, EUA, pag 173-185.

del proceso de optimización evolutiva. La situación antes descrita ocasiona que el algoritmo ocupe mucho de su tiempo computacional en sustituir los organismos enfermos. Esto a su vez, provoca que el proceso de exploración del espacio solución sea más lento y costoso¹⁵¹.

En sintonía con lo expuesto en el párrafo anterior, podemos citar a Premkumar, quién afirma que en general, la aplicación exitosa de los algoritmos genéticos está mucho más influenciada por el contexto restrictivo del problema que por su dimensionalidad¹⁵². A pesar de la interesante cantidad de investigaciones referentes al estudio analítico de los parámetros de un algoritmo genético, podemos afirmar que no existe un criterio de fijación general, permaneciendo mucho de su aplicación sobre el contexto particular del problema que esté siendo tratado así como de la empírica del investigador¹⁵³.

Aunque el objetivo del proyecto de investigación nunca fue establecer el conjunto de parámetros óptimos para el funcionamiento del algoritmo genético, por las razones ya antes expuestas, creemos que resulta importante al menos desarrollar un esfuerzo para su análisis predictivo. Lo anterior motiva al investigador a predecir aquellos niveles en los parámetros de operación del algoritmo genético, que pudieran ofrecer mejores expectativas de desempeño sobre la base particular del contexto restrictivo y dimensional del problema que está siendo atendido. Para dar respuesta a lo anterior, el investigador desarrollará a continuación un diseño experimental adicional.

La teoría del diseño de experimentos nos indica que para obtener resultados que sean concluyentes, debemos minimizar la cantidad de parámetros a ser manipulados en el experimento así como la cantidad de niveles para cada uno de los parámetros. El proceso anterior tiene el objetivo de minimizar la cantidad de combinaciones a ser revisadas lo cual equivale a minimizar la cantidad de tratamientos a ser aplicados en el experimento. Desafortunadamente, existen ocasiones en los que identificar a priori dichas combinaciones entre parámetros y sus niveles resulta inapropiado. En tal caso, lo que se requiere es establecer algún método que sistemáticamente vaya acotando el procedimiento de búsqueda combinatoria con la finalidad de lograr explorar de un modo eficiente el espacio solución.

¹⁵¹ Levine, G. (1997), Developing fitter genetic algorithms, *INFORMS Journal of Computing*, Vol. 9, No. 5, EUA, pag 251-253.

¹⁵² Premkumar, G. (1999), *Telecommunications Network design: a genetic algorithm approach*, School of Information Sciences and Technology & Pennsylvania State College of Business Administration, EUA, pag 19.

¹⁵³ Streifel, R. (1999), Dynamic Fussy control of genetic algorithm parameters, *IEEE Trans. Syst.*, EUA, pag 426-433.

En ese sentido de sistematización del diseño experimental para el análisis de los parámetros de un algoritmo genético, el principal esfuerzo jamás realizado en términos de investigación es lo hecho por Stewardson y Hicks. Ellos proponen el uso de un diseño de experimentos “esparcido” (sparse en inglés) basado en una estrategia secuencial de dos etapas. La primera etapa corresponde a una fase de “visualización” (screening en inglés) con la aplicación de una cantidad limitada de replicas, mediante la cual se identifican los niveles más alto y más bajo correspondientes a cada uno de los parámetros experimentados. Los resultados de esta primera fase son procesados a través de un modelo general de ANOVA con la finalidad de descartar los niveles de aquellos parámetros que estadísticamente resulten con menor significancia relativa. Con los resultados obtenidos mediante ANOVA, se procede a desarrollar una segunda fase de experimentación sobre la base de una menor cantidad de niveles entre los parámetros y una mayor cantidad de replicas que en la fase anterior. Lo anterior hace plausible que el poder estadístico de esta segunda fase secuencial sea mucho más concluyente con respecto a lo que se pueda obtener de la primera¹⁵⁴.

¹⁵⁴ Stewardson, D. (2002), *Overcoming Complexity: optimizing genetic algorithms for use in complex scheduling problems via designed experiments*, University of Newcastle, United Kingdom, pag 5-13.

9.5 Diseño experimental y resultados para la predicción de los parámetros del Algoritmo Genético.

Conviene recordar en este momento que el diseño experimental que fue planteado para dar respuesta a la hipótesis del proyecto de investigación, se desarrolló sobre la base de fijar convenientemente los parámetros principales del algoritmo genético de acuerdo a la empírica del investigador. Sobre la base de esta empírica, podemos proponer que los parámetros del algoritmo genético que sean experimentados para optimizar su calibración, son los siguientes:

1. Tamaño de la población de organismos (N)
2. Cantidad de generaciones (G)
3. Factor de cruzamiento (FC)
4. Factor de mutación (FM)

El diseño experimental a desarrollar deberá cubrir y balancear la cantidad de mediciones a ser aplicadas para cada uno de los parámetros (o también llamados predictores), así como de sus niveles entre si. Así entonces procederemos a ejecutar una serie de mediciones sobre la base combinatoria de estos 4 parámetros y convenientemente tomando en consideración 4 niveles para cada uno de estos. El diseño experimental será desarrollado a partir de 10 replicas para cada uno de los 16 tratamientos (4 niveles para cada parámetro) con la finalidad de incrementar el poder estadístico del experimento.

Las Tablas 9.7, 9.8 y 9.9 muestran el diseño experimental a ser desarrollado. La Tabla 9.7, muestra las combinaciones así como los significados cuantitativos, para cada uno de los 4 niveles con que operan los parámetros relacionados al tamaño de la población (N) y a la cantidad de generaciones (G).

Tabla 9.7 Diseño experimental para los parámetros N y G.

Tamaño de la Población (N)		Cantidad de Generaciones (G)				Valor Real
		5,000	10,000	20,000	40,000	
Valor Real	Val.Estd.	1	2	3	4	Val.Estd.
50	1	A1= X11	A2= X12	A3= X13	A4= X14	
100	2	A5= X21	A6= X22	A7= X23	A8= X24	
150	3	A9= X31	A10= X32	A11= X33	A12= X34	
200	4	A13= X41	A14= X42	A15= X43	A16= X44	

De la Tabla 9.7, podemos entonces ejemplificar al elemento $A10= X32$, como aquel tratamiento correspondiente a un tamaño de población (N) fijado a 150 organismos y con una cantidad de generaciones (G) fijado a un límite de 10,000 generaciones. A continuación abajo en la Tabla 9.8, se muestra las combinaciones así como los significados cuantitativos, para cada uno de los 4 niveles con que operan los parámetros relacionados al factor de cruzamiento (FC) y al factor de mutación (FM).

Tabla 9.8 Diseño experimental para los parámetros FC y FM.

Factor de Cruzamiento (FC)		Factor de Mutación (FM)				Valor Real
		2.5%	5%	7.5%	10%	Valor Real
Valor Real	Val.Estd.	1	2	3	4	Val.Estd.
20%	1	B1= Y11	B2= Y12	B3= Y13	B4= Y14	
40%	2	B5= Y21	B6= Y22	B7= Y23	B8= Y24	
60%	3	B9= Y31	B10= Y32	B11= Y33	B12= Y34	
80%	4	B13= Y41	B14= Y42	B15= Y43	B16= Y44	

De la Tabla 9.8, podemos entonces ejemplificar al elemento $B14= Y42$, como aquel tratamiento correspondiente a un factor de cruzamiento (FC) fijado al 80% y con un factor de mutación (FM) fijado a un 5%. Finalmente en la Tabla 9.9, se muestra las combinaciones para cada uno de los 4 niveles con que operan los 4 parámetros involucrados en el experimento (N, G, FC y FM). Esto equivale entonces a un total de 256 combinaciones posibles, es decir 4 parámetros combinados a 4 niveles ($4^4 = 256$).

Tabla 9.9 Diseño experimental para los parámetros N, G, FC y FM.

INTERACCION N-G		INTERACCION FC-FM					Valor Real
		Y11	Y12	Y43	Y44	Valor Real
Valor Real	Val.Estd.	B1	B2	B15	B16	Val.Estd.
X11	A1	T1= X11,Y11	T2= X11,Y12	T15= X11,Y43	T16= X11,Y44	
X12	A2	T17= X12,Y11	T18= X12,Y12	T31= X12,Y43	T32= X12,Y44	
.....	
X43	A15	T225= X43,Y11	T226= X43,Y12	T239= X43,Y43	T240= X43,Y44	
X44	A16	T241= X44,Y11	T242= X44,Y12	T255= X44,Y43	T256= X44,Y44	

Sobre la base de un total de 256 tratamientos y una cantidad de 10 replicas, se tiene entonces un total de 2560 ejecuciones del algoritmo genético. Dicha información estadística es transferida al software computacional SPSS ver 11.0 para su análisis correspondiente. A continuación en la Tabla 9.10, se muestran los resultados provenientes del análisis de varianza:

Tabla 9.10 Análisis de Varianza Univariado: Nivel de significancia para los predictores.

Dependent Variable: FOBJ

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	2044506.597 ^a	4	511126.649	3268.080	.000
Intercept	1759480.821	1	1759480.821	11249.901	.000
TPOB	2040286.114	1	2040286.114	13045.335	.000
NGENE	976.483	1	976.483	6.244	.013
FCRUCE	11.357	1	11.357	.073	.788
FMUTA	619.156	1	619.156	3.959	.047
Error	399601.163	2555	156.400		
Total	10134069.1	2560			
Corrected Total	2444107.760	2559			

a. R Squared = .937 (Adjusted R Squared = .936)

El valor del estadístico “F” (penúltima columna) es calculado dividiendo el promedio cuadrado de cada parámetro (columna 4) entre el promedio cuadrado del error (619,15). El nivel de significancia (o valor de “p”) para cada parámetro, es verificado en la última columna. En general, un parámetro con un valor de “p” por abajo del 5% se le considera como significativo. De los coeficientes de la prueba “F” (penúltima columna) y de los valores de significancia o valores “p” (última columna), revisados en la tabla anterior se pueden obtener los siguientes resultados:

1. El parámetro “TPOB”, que se refiere al tamaño de la población (N), es el que obtiene el mayor nivel de significancia (1-p = 100%) de los 4 predictores que están siendo confrontados, lo que significa que el tamaño de la población (N) influye decididamente en el valor de la función objetivo obtenido por el algoritmo genético.
2. Los siguientes 2 parámetros de mayor significancia en ese mismo orden, son “NGENE” y “FMUTA”. El primero se refiere a la cantidad de generaciones (G) ejecutadas por el algoritmo genético el cual obtiene un nivel de significancia bastante alto (1-p = 98.7%). El segundo parámetro se refiere al factor de mutación (FM) que es empleado por el algoritmo genético para

mantener la diversidad en la población de organismos. En este caso obtenemos un nivel de significancia suficiente para las expectativas del investigador ($1-p=95.3\%$)

3. En lo concerniente al último parámetro FCRUCE el cual corresponde al factor de cruzamiento (FC) utilizado por el algoritmo genético durante el proceso de reproducción de los organismos, tenemos que su valor de “p” excede del 5%, por lo que entonces podemos deducir que dicho parámetro no establece fuertemente una relación causa-efecto sobre la función objetivo obtenida en el algoritmo genético.
4. El valor de R^2 ajustado es de 93.6%. Al valor anterior se le conoce como coeficiente de determinación el cual se le utiliza como una métrica de bondad de ajuste respecto al modelo lineal. Dicho de otro modo, éste valor puede ser entendido como la proporción de la variabilidad de la variable dependiente (en nuestro caso es el valor de la función objetivo) que puede ser explicada a través del modelo de regresión que está siendo utilizado. Un porcentaje de determinación pequeño, significaría que el modelo no se ajusta adecuadamente a los datos. Podemos finalmente decir que, el coeficiente de determinación obtenido en nuestro caso satisface plenamente los requerimientos del experimento.

Finalmente a continuación en la Tabla 9.11, se muestran los resultados para la estimación de los coeficientes de predicción para cada uno de los parámetros. Cabe mencionar que el cálculo de dichos coeficientes se realizó sobre la base de un modelo lineal de predicción el cual se encuentra apoyado sobre la base del coeficiente de determinación del 93.6% obtenido en el análisis anterior.

Tabla 9.11 Estimación estadística de los coeficientes e intervalos de confianza para los parámetros de operación del algoritmo genético.

Dependent Variable: FOBJ

Parameter	B	Std. Error	t	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
Intercept	119.064	1.123	106.066	.000	116.863	121.265
TPOB	-24.923	.218	-114.216	.000	-25.351	-24.495
NGENE	-.549	.220	-2.499	.013	-.979	-.118
FCRUCE	5.957E-02	.221	.269	.788	-.374	.493
FMUTA	-.435	.219	-1.990	.047	-.863	-6.290E-03

La 2da. columna de la tabla anterior, corresponde al valor del coeficiente del modelo lineal que está siendo estimado para cada uno de los parámetros que aparecen a su vez en la 1era. columna. La 4ta. columna corresponde al valor de la prueba “t” de student, el cual se calcula dividiendo el valor del coeficiente de cada parámetro entre su error estándar. En términos generales la prueba “t” de student, se utiliza para determinar el grado de importancia relativa que cada uno de los parámetros guarda en el modelo. Como regla general podemos establecer que aquellos parámetros cuyo valor absoluto de “t” sea mayor o igual a 2, resultan ser buenos predictores respecto a la variable dependiente.

El valor de “t” antes explicado a su vez, se asocia con el nivel de significancia (valor de “p”) el cual se reporta en la 5ta. columna. Dicho valor guarda una interpretación equivalente a la reportada en la prueba “F” de análisis de varianza de la Tabla 9.10 ya antes expuesta. Finalmente los valores que aparecen en la 6ta y 7ma. columna corresponden a los intervalos de confianza al 95% para los coeficientes del modelo lineal calculados para cada parámetro en la columna 2. De los coeficientes reportados en la Tabla 9.11 podemos interpretar los siguientes resultados:

1. El parámetro “TPOB”, o tamaño de la población (N), tiene un coeficiente de estimación negativo (-24.923), lo cual significa una proporcionalidad inversa. Lo anterior comprueba que a medida que la población de organismos en el algoritmo genético se incrementa, en esa medida el valor de la función objetivo (costo) se logra disminuir en términos numéricos, lo cual debe ser entendido como un efecto favorable en la calidad de la solución. También podemos comprobar que el valor absoluto del coeficiente es el más grande que se reporta en el modelo, lo cual quiere decir que es el que también logra mayormente influir en el resultado de la función objetivo del algoritmo genético. Lo anterior se comprueba por igual al revisar el coeficiente “t” el cual en términos absolutos, también resulta ser el mayor en todo el modelo.
2. El parámetro “NGENE”, que se refiere a la cantidad de generaciones (G), tiene un coeficiente de estimación mucho menor en términos absolutos (-0.549), lo cual significa que aunque, a medida que transcurren las generaciones en el algoritmo genético se mejora el resultado de la función objetivo, no obstante, el efecto no resulta ser tan impactante en el resultado. De cualquier manera al revisar su coeficiente “t” (-2.499), éste nos sugiere que el parámetro no debe ser excluido del modelo.

3. El parámetro “FMUTA”, que se refiere al factor de mutación (FM), tiene un coeficiente de estimación muy comparable (-0.435) al revisado en el caso de la cantidad de generaciones (G). Esto nuevamente hace deducir a mayor factor de mutación en el algoritmo genético también se obtendrán mejores soluciones en la función objetivo. El valor de su coeficiente “t” (-1.990), resulta apenas ser suficiente para que el parámetro pueda catalogarse como relevante en el modelo de predicción. Lo que si resulta interesante observar en este parámetro respecto a lo verificado en los otros dos previamente expuestos, es en lo referente a los intervalos de confianza que aparecen registrados. El límite inferior del coeficiente (lower bound), favorece la hipótesis de que cualquier incremento en el factor de mutación mejorará la solución obtenida por el algoritmo genético. No obstante el valor de (-6.290 E-03) en el límite superior del coeficiente (upper bound), nos hace dudar de lo anterior sobre la base del intervalo de confianza fijado en el experimento al 95%. Este coeficiente (-0.000629) es tan pequeño, que bajo estas circunstancias, la influencia que el factor de mutación pudiera ejercer en la solución obtenida por el algoritmo genético sería muy marginal.

4. Finalmente el parámetro “FCRUCES”, que se refiere al factor de cruzamiento (FC), tiene el coeficiente de estimación en valor absoluto más pequeño de todos (5.957 E-02). Lo que de este valor podemos interpretar, es que dicho parámetro afecta marginalmente el resultado obtenido en la solución del algoritmo genético. El valor de su coeficiente “t” (-0.269), comprueba lo anterior. Nuevamente resulta interesante observar en este parámetro lo referente a los intervalos de confianza que aparecen registrados. El límite inferior del coeficiente es negativo (-0.374), mientras que el valor de su límite superior es positivo (0.493). Esta ambivalencia lo que nos hace deducir con respecto a este parámetro, es que al nivel de confiabilidad del 95% que había sido pretendido, no es posible probar la hipótesis que indique la medida en la que este parámetro afecta a la solución del algoritmo genético. Lo anterior no significa que debemos aceptar la hipótesis de que el factor de cruzamiento no tiene influencia en la solución del algoritmo genético, sino que para probarlo o rechazarlo se debería utilizar algún otro diseño experimental lo cual escapa al alcance y objetivo trazado en nuestro proyecto de investigación.

Con lo expuesto hasta aquí, establecemos entonces el siguiente modelo lineal de predicción para el comportamiento de la función objetivo solucionada a través del algoritmo genético propuesto:

$$F_{Obj.Est} = 119.064 - 24.923 tpob - 0.549 n_{gene} - 0.435 f_{muta}$$

Aunque el modelo lineal de predicción tiene un coeficiente de determinación $R^2_{ajustado}$ igual al **93.6%** (revisar tabla 9.10), el cual es bastante razonable, no obstante, el investigador a continuación propone un modelo de regresión No-Lineal con el propósito de finalmente estimar la brecha de optimalidad en el poliedro del problema SPDP-sTW, el cual a su vez esta descrito a través del diseño factorial que fue aplicado a los parámetros de operación del algoritmo genético propuesto.

El modelo de regresión NO-lineal que a continuación se propone está compuesto por 3 componentes polinomiales para cada uno de los parámetros de operación del algoritmo genético. Los 3 componentes del polinomio son: uno lineal, otro cuadrático y finalmente uno más del tipo hiperbólico. A continuación se expone el modelo propuesto:

$$F_{Obj.Est} = A_0 + A_1 * tpob + A_2 * tpob^2 + \frac{A_3}{tpob} + B_1 * ngene + B_2 * ngene^2 + \frac{B_3}{ngene} + C_1 * fcruza + C_2 * fcruza^2 + \frac{C_3}{fcruza} + D_1 * fmuta + D_2 * fmuta^2 + \frac{D_3}{fmuta}$$

El modelo anterior fue resuelto mediante la minimización de la suma de diferencias cuadráticas a través de un algoritmo convencional de programación lineal. El coeficiente de determinación $R^2_{ajustado}$ arrojó **98.2%**, lo cual supera al obtenido a través del modelo de predicción lineal. A continuación en la Tabla 9.12, se presentan los coeficientes obtenidos para el modelo:

Tabla 9.12 Coeficientes obtenidos para el modelo polinomial.

PARAMETROS DEL MODELO	X	X^2	1/X	PARAM. OPT
	LINEAL	CUADRÁTICO	HIPERBÓLICO	
TPOB (Ai)	2.1865	-0.2581	106.1476	4.0
NGENE (Bi)	-1.6492	0.1966	-1.2288	4.0
FCRUCE (Ci)	-1.3911	0.1740	-2.0155	3.5348
FMUTA (Di)	-6.6890	1.0300	-4.3781	3.0130
A0	13.2691			

Las columnas 2, 3 y 4 corresponden a la exposición de los coeficientes del modelo polinomial antes descrito. La columna 2 corresponde al término lineal. La siguiente columna al término cuadrático y la cuarta al término hiperbólico. Finalmente la última columna corresponde a los valores óptimos calculados para cada uno de los parámetros del algoritmo genético descritos a través del modelo polinomial antes mencionado. Existen básicamente dos métodos para calcular los valores óptimos para los parámetros del modelo:

1. Aplicando al modelo polinomial 4 derivadas parciales. Cada derivada parcial corresponde a cada uno de los parámetros del algoritmo genético. Cada derivada parcial a su vez, es igualada a cero (gradiente igual a cero) para determinar el punto de inflexión de la curva de costo de la función objetivo con respecto al parámetro en cuestión. De esta manera se obtienen 4 ecuaciones con 4 incógnitas (corresponden a los 4 parámetros del modelo polinomial), las cuales pueden ser resueltas por cualquier método de convergencia.
2. Aplicando cualquier algoritmo de optimización que pueda resolver problemas cuya superficie de respuesta NO esté delimitada por hiper-planos lineales. Naturalmente, este tipo de problemas no puede ser atendido a través de métodos de solución exacta sino solo a través de métodos de aproximación.

En el caso nuestro el investigador no dejó pasar la oportunidad para volver a aplicar, una vez más, el algoritmo genético propuesto para aproximar los valores óptimos correspondientes a los parámetros del modelo polinomial. Así entonces, si sustituyéramos los valores óptimos calculados en el modelo de regresión polinomial, obtendríamos entonces un valor mínimo en el costo de la función objetivo equivalente a “25.094”, el cual es el valor que utilizaremos como cota inferior para estimar la brecha de optimalidad en el poliedro del problema SPDP-sTW.

Hecho lo anterior, ya solo faltaría calcular un valor apropiado para el costo de la función objetivo sobre la cual, al compararla versus la cota inferior, dicha diferencia cuantitativa fuese entonces interpretada analíticamente como la brecha de optimalidad a poder potenciar como beneficio económico. Para poder establecer dicho costo referencial analizaremos a continuación en la Tabla 9.13, la Distribución estadística del costo en la función objetivo para el espacio solución.

Tabla 9.13 Distribución estadística del costo en la función objetivo para el espacio solución.

Distribución Estadística del Costo en la Función Objetivo para el Espacio Solución							
Suma Aritmética de los Parámetros	Rangos de Valores obtenidos en el Costo de la Función Objetivo						TOTALES
	25-30	30-35	35-40	50-55	60-105	105-110	
4	0.0%	0.0%	0.0%	0.0%	0.0%	0.4%	0.4%
5	0.0%	0.0%	0.0%	0.5%	0.7%	0.4%	1.6%
6	0.0%	0.0%	0.3%	1.3%	1.6%	0.7%	3.9%
7	0.2%	0.0%	1.2%	2.0%	4.1%	0.5%	7.9%
8	1.5%	0.5%	1.8%	4.1%	4.7%	0.0%	12.5%
9	2.4%	0.9%	2.7%	4.6%	4.9%	0.0%	15.4%
10	3.5%	1.8%	2.9%	4.7%	4.3%	0.0%	17.1%
11	4.3%	2.4%	1.9%	3.8%	2.5%	0.0%	15.0%
12	5.3%	2.9%	0.9%	2.4%	1.1%	0.0%	12.5%
13	3.8%	2.0%	0.5%	1.1%	0.3%	0.0%	7.7%
14	2.5%	0.9%	0.0%	0.3%	0.0%	0.0%	3.8%
15	1.7%	0.3%	0.0%	0.0%	0.0%	0.0%	2.0%
16	0.4%	0.0%	0.0%	0.0%	0.0%	0.0%	0.4%
TOTALES	25.5%	11.6%	12.0%	24.6%	24.3%	2.0%	100.0%

En la Figura 9.1 se puede comprobar la normalidad que existe en la distribución de frecuencias correspondientes al diseño factorial que fue empleado para el experimento.

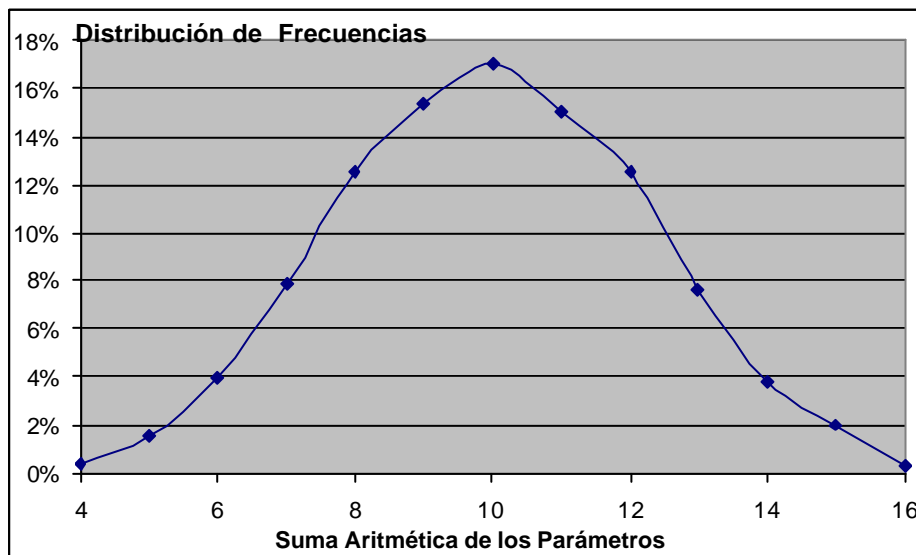


Figura 9.1 Distribución de frecuencias para la suma aritmética de parámetros del algoritmo genético.

Finalmente en la Figura 9.2, se presenta la Distribución Acumulada del espectro del Espacio Solución del problema SPDP-sTW descrito por los parámetros del algoritmo genético.

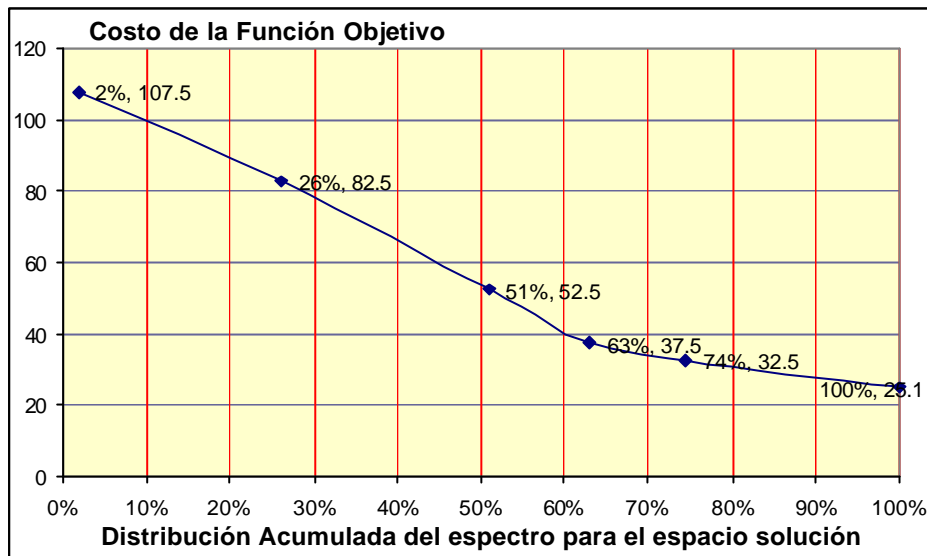


Figura 9.2 Distribución Acumulada del espectro del Espacio Solución del problema.

Para explicar la interpretación de la curva anterior podemos dar el siguiente ejemplo. Más del 40% del espacio solución del problema oscila apenas entre un valor del costo aproximado a 40 y el valor mínimo de 25.1, identificado ya anteriormente como el valor óptimo esperado (**25.094**). Dicho de otro modo el 40% de las soluciones del espacio total, apenas logran mejorar el porcentaje de optimalidad en un 13.9% al disminuir 14.9 puntos el valor del costo en la función objetivo con respecto a la cota superior (107.5), el cual es calculado a través de la curva descrita por el modelo polinomial.

De la Figura 9.2, podemos establecer que una adecuada referencia contra la cual comparar la cota inferior para determinar la brecha de optimalidad, pudiera ser a través del valor promedio del costo en la función objetivo descrito por la curva del modelo polinomial. Dicho costo promedio estaría ponderado por el tamaño del espacio solución en la función objetivo del problema. De la Figura 9.2, podemos verificar que dicho costo promedio equivale a 52.9 aproximadamente. Entonces la brecha de optimalidad puede ser finalmente calculada de la siguiente manera:

$$\frac{52.9 - 25.094}{107.5} \approx 25.87\%$$

Como puede ser revisado, éste valor estimado supera el 10% y el 25% que habíamos calculado mediante el muestro en el apartado 9.3.

9.6 Significancia de los parámetros del Algoritmo Genético en relación a su eficacia.

Con los análisis tratados hasta este momento, ya solo queda una interrogante que el investigador busca atender. Dicha cuestión consiste en estimar el grado de mejoramiento que puede alcanzarse en el desempeño del algoritmo genético, al modificar los valores que fueron empíricamente establecidos al realizar el diseño experimental para probar la hipótesis de investigación. Hemos de recordar que dicho experimento fue diseñado en el capítulo # 8 y sus resultados fueron expuestos en el apartado 9.2 del presente capítulo.

El presente análisis no pretende sustituir los hallazgos y la hipótesis del investigador respecto a la eficacia con la cual el algoritmo genético se enfrenta al grupo control. El razonamiento a lo antes mencionado, se tiene al revisar el apartado 9.5 en el cual se expuso el modelo lineal de predicción. En dicho modelo de predicción, se establece que el algoritmo genético alcanza un valor en la función objetivo más favorable al incrementar el tamaño de la población. No obstante en dicho análisis, no se toma en consideración el parámetro del tiempo computacional, el cual resulta ser fundamental en el diseño experimental desarrollado en el apartado 9.2 para la prueba de la hipótesis de investigación. Así entonces, es comprensible que ambos diseños experimentales buscan objetivos diferentes.

Lo que buscamos ahora atender es relacionar los parámetros del algoritmo genético no con respecto al valor de la función objetivo por sí sola, como fue desarrollado en el apartado 9.5, sino con respecto a la eficacia con la cual se logra el valor de la función objetivo. Dicha métrica de eficacia se refiere precisamente a la combinación de los dos parámetros que fueron aplicados en el diseño experimental para la prueba de la hipótesis expuesta en el apartado 9.2, es decir, el porcentaje de optimalidad y el tiempo computacional.

De lo expuesto en el apartado 9.5, podemos descartar al parámetro FRCUZA (factor de cruzamiento), ya que su influencia en el desempeño del algoritmo genético, carece de significancia estadística según los resultados obtenidos en el experimento. En lo referente al parámetro NGENE (número de generaciones), aunque en el apartado 9.5 se evidencia como un parámetro significativo con respecto al valor de la función objetivo, no obstante es fácil darse cuenta que su valor es directamente proporcional al parámetro del tiempo computacional. Dicho de otro modo, ambos parámetros miden lo mismo, por lo que no es posible entonces relacionarlos.

De lo anterior, podemos entonces identificar solo dos parámetros restantes que son el tamaño de la población (TPOB) y el factor de mutación (FMUTA). Por tanto, el análisis a desarrollar consiste en estimar la relación de estos dos parámetros del algoritmo genético con respecto a la eficacia en el porcentaje de optimalidad alcanzado. Aunque en el apartado 9.5 se comprueba que ambos parámetros son significativos con respecto al valor de la función objetivo, no obstante, no es posible afirmar lo mismo con respecto a la eficacia en el porcentaje de optimalidad alcanzado. Para este análisis se utilizarán los mismos datos que se aplicaron en el apartado 9.5 con la excepción de que se añadirá un parámetro adicional que es el que se refiere al tiempo computacional (TIEMPO). Dicho tiempo computacional corresponde a aquel que fue registrado durante el levantamiento de los datos que fueron ocupados para el experimento expuesto en el apartado 9.5.

La nueva variable que será utilizada para medir la eficacia en el porcentaje de optimalidad alcanzado, la denominaremos simplemente como (EFICACIA). Como es de esperarse, la forma como mediremos la variable EFICACIA será a través de la interacción del valor en la función objetivo (FOBJ) obtenido mediante el algoritmo genético, junto con el valor del tiempo computacional registrado (TIEMPO). Así entonces la formulación que usaremos para calcular la variable EFICACIA es la siguiente:

$$EFICACIA_i = \sqrt{\frac{FOBJ_{opt} * TIEMPO_{min}}{FOBJ_i * TIEMPO_i}} \quad ? \quad i = 1 .. n \quad \text{donde } n = 1..2560$$

$$FOBJ_{opt} = 25.096$$

$$TIEMPO_{min} = \min(TIEMPO_i)$$

Como puede ser visto en la formulación anterior, la variable EFICACIA mide el desempeño del algoritmo genético simultáneamente en dos dimensiones, es decir, con respecto al valor óptimo esperado de la función objetivo y con respecto a un valor referencial de tiempo. Dicha referencia de tiempo, convenientemente se fija al identificar el tiempo mínimo que haya sido registrado en el experimento. Así entonces, la variable EFICACIA mide a la vez dos aspectos:

1. Que tan lejos queda el valor de la función objetivo respecto al valor óptimo esperado.
2. Con que rapidez se obtiene dicho valor en la función objetivo

Ambos factores de desempeño se multiplican entre sí, con la finalidad de calcular el efecto conjunto que es el que realmente nos interesa medir. La raíz cuadrada en la formula, simplemente se utiliza para obtener el desempeño promedio de ambos factores conjuntamente.

Hecho lo anterior, ejecutamos el análisis de significancia en el software estadístico SPSS Ver 11.0 con la finalidad de verificar los resultados. A continuación se muestra en la Tabla 9.14 el análisis de varianza.

Tabla 9.14 Nivel de significancia para los predictores de la variable Eficacia.

Tests of Between-Subjects Effects

Dependent Variable: EFICACIA

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	23.102 ^a	4	5.776	1594.384	.000
Intercept	3.609	1	3.609	996.381	.000
TPOB	6.597E-03	1	6.597E-03	1.821	.177
FMUTA	.511	1	.511	141.113	.000
FOBJ	3.871E-02	1	3.871E-02	10.687	.001
TIEMPO	3.638	1	3.638	1004.272	.000
Error	9.255	2555	3.622E-03		
Total	270.664	2560			
Corrected Total	32.357	2559			

a. R Squared = .814 (Adjusted R Squared = .814)

De la tabla anterior, podemos destacar los siguientes resultados:

1. El valor de la función objetivo (FOBJ) así como el tiempo computacional aplicado (TIEMPO), ambos elementos resultan ser significativos al calcular la eficacia en el desempeño del algoritmo genético (EFICACIA). La variable EFICACIA debe ser entendida como la propiedad activa en el algoritmo genético para lograr obtener soluciones de buena calidad en tiempos computacionales pequeños (calidad vs tiempo).
2. El parámetro “Tamaño de la población” (TPOB), no es significativo al 5% de confiabilidad (1-p = 82.3%), como predictor de la eficacia en el desempeño del algoritmo genético. Lo anterior comprueba que aunque, el manejar una población grande de individuos contribuye a aumentar las posibilidades de obtener mejores soluciones debido a la diversidad de material genético presente en la población, no obstante, el costo incurrido para sostener una población más grande se repercute en el tiempo computacional para su procesamiento.

3. El “factor de mutación” (FMUTA), si resulta ser un componente significativo en lo referente al balance entre la obtención de soluciones de buena calidad versus el tiempo (ó costo) requerido para lograrlo. Lo anterior, no significa que al aumentar el valor del parámetro FMUTA, necesariamente se obtendrá un mejor desempeño en la variable EFICACIA. Lo que quiere decir, es que el valor de FMUTA, tiene un rol definitivamente significativo en el desempeño del algoritmo genético al hablar del binomio (calidad vs tiempo).
4. El coeficiente de determinación ($R^2_{ajustado}$) es del 81.4%. Esto quiere decir que la proporción de la variabilidad registrada en la variable EFICACIA puede ser explicada en un 81.4% a través de los predictores (TPOB y FMUTA). Aunque el coeficiente es razonablemente bueno, no obstante, lo anterior hace prever que deben existir otros elementos que están influenciando el resultado de la eficacia en el desempeño del algoritmo genético. El tratamiento para el análisis de dicha fuente de variabilidad está fuera del alcance de la presente tesis. No obstante, el investigador sugiere que dicha variabilidad está originada primordialmente por el manejo del resto de las fases algorítmicas (heurísticas) que están siendo incluidas en adición al propio algoritmo genético¹⁵⁵.

A continuación se muestra en la Tabla 9.15 el cálculo de los coeficientes de estimación así como de los intervalos de confianza.

Tabla 9.15 Coeficientes de estimación e intervalos de confianza.

Parameter Estimates

Dependent Variable: EFICACIA

Parameter	B	Std. Error	t	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
Intercept	.413	.013	31.566	.000	.388	.439
TPOB	6.006E-03	.004	1.349	.177	-2.721E-03	1.473E-02
FMUTA	1.415E-02	.001	11.879	.000	1.181E-02	1.648E-02
FOBJ	-3.41E-04	.000	-3.269	.001	-5.457E-04	-1.365E-04
TIEMPO	-9.34E-03	.000	-31.690	.000	-9.922E-03	-8.766E-03

¹⁵⁵ Supra: revisar el apartado 7.7

De la tabla anterior, podemos destacar los siguientes resultados:

1. El coeficiente de TPOB ($6.006E-03$), evidencia que dicho parámetro ofrece una influencia predictiva muy baja sobre el valor numérico de la variable EFICACIA.
2. El intervalo de confianza obtenido para el coeficiente TPOB ($-2.721E-03$ a $1.473E-02$) deja claro que la influencia de dicho parámetro en la variable EFICACIA, aparte de ser pequeña, también es ambigua. Lo anterior se explica por el cambio de signo que existe entre el intervalo inferior y el superior del coeficiente.
3. Aunque los parámetros FOBJ y TIEMPO tienen coeficientes muy pequeños, esto es previsible, ya que éstos fueron utilizados para calcular directamente el valor de la variable EFICACIA.
4. Los valores de ambos coeficientes (FOBJ y TIEMPO) son de signo negativo, por lo que se confirma la tendencia como ambos parámetros influyen el valor de la variable EFICACIA. Lo anterior quiere decir, que en ambos parámetros la afectación es inversamente proporcional, es decir, a menor valor en la función objetivo (o sea mejor calidad) y a menor tiempo registrado (más rápido), entonces la eficacia del algoritmo aumenta. Los signos reportados en los intervalos de confianza (superior e inferior) no dejan ninguna sospecha al respecto.
5. El coeficiente del parámetro relacionado al factor de mutación (FMUTA), es el que ofrece en términos relativos, la mayor influencia ($1.415E-02$) sobre la eficacia en el desempeño del algoritmo genético. Los signos reportados en el intervalo de confianza, también muestran que este parámetro mantiene una tendencia consistente en relación a la variable EFICACIA.
6. El coeficiente del parámetro (FMUTA) es positivo, lo cual significa que la tendencia es directamente proporcional con respecto a la variable EFICACIA. Esto quiere decir que sería previsible que al aumentar el factor de mutación en el algoritmo genético, también fuese esperable que aumentase la eficacia del algoritmo genético. El valor del coeficiente indica un gradiente del 1.42% en el mejoramiento de la eficacia con respecto al factor de mutación
7. A pesar del valor del gradiente anterior, no es posible concluir que por cada incremento del 1% en el factor de mutación fuese posible indefinidamente obtener un mejoramiento del 1.42% en la eficiencia del algoritmo genético. El razonamiento anterior es sobre la base de que el modelo de predicción que fue aplicado para el análisis de significancia es lineal y por tanto solo es posible establecer un criterio de tendencia y no de optimalidad como ocurrió con el modelo polinomial desarrollado en la tabla 9.12 ya antes expuesto.

Con el análisis anterior, damos por concluido el presente capítulo quedando satisfechos con los resultados expuestos.

9.7 Comentarios finales.

Con la exposición anterior, damos por terminado el capítulo referente a los resultados obtenidos en la fase de experimentación. Como pudimos comprobar, el diseño experimental expuesto, dio tratamiento a los siguientes aspectos:

1. En el apartado 9.2 se dio respuesta a la hipótesis de investigación mediante un análisis comparativo de las soluciones obtenidas mediante el algoritmo genético propuesto con respecto al grupo control. El grupo control fue constituido de las soluciones óptimas verificables obtenidas mediante el uso de un algoritmo de ramificación y corte.
2. En el apartado 9.3, se desarrolló un análisis costo beneficio. Este contribuyó a ofrecer una justificación económica acerca de la viabilidad del algoritmo propuesto para mejorar la capacidad de transporte y el servicio al cliente sobre la base de supuestos conservadores definidos empíricamente por el investigador.
3. En el apartado 9.4 se establecieron las bases para el desarrollo de un modelo de regresión lineal con la finalidad de determinar la significancia estadística así como la importancia relativa que cada uno de los parámetros del algoritmo genético tienen con respecto a la obtención de mejores soluciones en la función objetivo.
4. A partir del modelo de regresión lineal anterior, en el apartado 9.5 se propone un modelo de regresión polinomial. Mediante éste último se comprueban bs supuestos de mejora en la métrica de optimalidad que fueron propuestos por el investigador en el apartado 9.3. El modelo de predicción permite estimar el tamaño de la brecha de optimalidad. Los resultados del modelo analítico sobre pasan los supuestos de mejora establecidos en el apartado 9.3.
5. En el apartado 9.6, se desarrolla un análisis para determinar la significancia de los parámetros del algoritmo genético con respecto a su eficacia en el tratamiento del binomio “calidad vs tiempo”. Los resultados muestran que aunque el tamaño de la población si resulta ser significativo en cuanto al valor de la función objetivo, no obstante, el tiempo computacional queda comprometido.

Finalmente, es importante destacar que para poder establecer conjeturas definitivas acerca de la definición de los niveles óptimos de operación para los parámetros involucrados en el algoritmo genético propuesto, sería necesario desarrollar estudios y diseños experimentales particulares tales como los expuestos en el apartado 9.4 desarrollados por Stewardson y Hicks.

10. Conclusiones.

I. En lo referente a la Filosofía de la Ciencia de la Administración:

1. La Ciencia de la Administración (OR/MS) es comunicada a través de la modelación mediante relaciones lógicas las cuales proveen justificaciones verificables para las inferencias producidas.
2. A pesar de esto, lo anterior no contribuye a evidenciar el proceso psicológico mediante el cual las inferencias son desarrolladas por primera ocasión lo cual permanece siendo básicamente intuitivo, ya que el proceso para abstraer un problema administrativo en uno científico sigue siendo un arte.
3. Una buena práctica en la modelación matemática en el OR/MS es aplicando un enfoque evolutivo mediante una secuencia de modelos de alcance y complejidad incremental. Deliberadamente se omiten temporalmente consideraciones del problema. Si el modelo se mantiene tratable entonces se continua el proceso de enriquecimiento de otro modo se modifican y/o simplifican los supuestos.
4. La clave en la aplicación del OR/MS es la identificación de aquellas consideraciones que permitan abstraer una realidad y que a la vez mantengan su solución computacional de manera tratable.

II. En lo referente a la aplicación práctica de la Ciencia de la Administración:

1. El aprendizaje del OR/MS implica el ejercicio de un pensamiento activo y crítico mediante el cual se vincula el conocimiento nuevo con el previo mediante una relación no memorizada, sino construida, contribuyendo al desarrollo de abstracciones con un significado tangible que es posible transferir a diversas situaciones para solucionar nuevos problemas.
2. En pocos profesionistas se puede afirmar que el OR/MS ha jugado un rol preponderante en sus carreras profesionales. Menos del 6% de los profesionistas de las empresas del AMM aplican las matemáticas en la toma de decisiones en general.
3. Existe una falta de aplicación del OR/MS en la administración de las empresas en general y en la logística en particular. Solo 2 empresas de las 55 muestreadas (menos del 5%) afirmaron utilizar como estrategia de solución para sus problemas de logística el desarrollo de modelos matemáticos.
4. La extrapolación del pasado es la principal herramienta para la toma de decisiones logística en las empresas del AMM. El 32% de las empresas muestreas del AMM aplican tan solo la experiencia.

III. En lo referente a la oferta de software comercial disponible actualmente para las empresas:

1. La problemática de la logística actualmente está siendo atendida por dos habilitadores: (1) el OR/MS y (2) la oferta de software computacional (sistemas ERP's y APS's).
2. Las investigaciones realizadas indican la falta de rigor matemático de los métodos de optimización encontrados en la oferta comercial de software APS. Solo se incluyen heurísticas poco sofisticadas.
3. El análisis comparativo referente a la oferta de software comercial APS para atender nuestro problema de investigación, evidencia lo improbable que resulta el que una sola alternativa de software "genérico" logre incluir todos los aspectos particulares que requieren ser considerados para los diversos escenarios a tener que ser considerados en la logística.
4. Tan solo 7 de las 55 empresas muestreadas del AMM han intentado resolver el problema de ruteo a través de la implementación de sistemas APS's. No reportan éxito a cabalidad.
5. Mientras las empresas no asocien y a la vez diferencien el uso de los sistemas ERP/APS con la aplicación del OR/MS, habrá un pobre aprovechamiento de las inversiones hechas en los primeros.
6. La aplicación del OR/MS puede mitigar la falta de funcionalidad encontrada en los sistemas APS.

IV. En lo referente al planteamiento del problema y a la estrategia de solución propuesta:

1. De las 55 empresas muestreadas del AMM, casi el 20% enfrentan variantes similares a la tratada como objetivo en nuestro proyecto de investigación relacionada a la logística de ruteo.
2. El problema planteado usualmente en la práctica requiere ser atendido con instancias de gran escala (=70 clientes) y con un fuerte porcentaje de ventanas de horario activas (=90%). Además las estructuras de amplitud de las ventanas de horario también es muy alto (=75%) lo cual evidencia el innegable compromiso real en el tema del servicio al cliente.
3. Nuestro problema de investigación es NP-Hard y apunta hacia una de las más ricas clases de problemas combinatorios en el OR/MS. La variante del problema está diferenciada de investigaciones previas a la vez que no existen extensamente aplicaciones de algoritmos genéticos.
4. La complejidad matemática intrínseca en el planteamiento de nuestro problema de investigación hace prever la necesidad de desarrollar algoritmos de solución que puedan ofrecer soluciones razonablemente buenas en tiempos de ejecución computacional cortos.
5. Aunque nuestro problema parte de una circunstancia orientada a la actividad humana (ciencias sociales), nuestra propuesta de investigación hizo uso de las ciencias exactas, las ciencias naturales y las ciencias computacionales. Los resultados son satisfactorios.

V. En lo referente a los resultados obtenidos en el experimento para la prueba de la hipótesis:

1. El Algoritmo de Ramificación y Corte BC que se aplica como instrumento para el Grupo Control en el diseño experimental, explota eficientemente la estructura matemática del problema logrando alcanzar la solución óptima para las instancias del problema descritas como “sencillas”, las cuales tienen características comparables a las que Ascheuer, Fischetti, y Grottschel exponen en su investigación (2001).
2. Además, nuestro Algoritmo BC obtiene soluciones “al menos” muy cercanas al óptimo para las instancias que son particularmente difíciles de resolver y en las cuales el proyecto de investigación se concentra.
3. La hipótesis del proyecto de investigación es comprobada:
Podemos establecer a un 100% de confiabilidad que el algoritmo genético propuesto alcanza un porcentaje de optimalidad > 90% en un tiempo computacional = 5 minutos.
4. También podemos establecer que nuestra versión avanzada del algoritmo genético (propuesto) ofrece soluciones al problema de investigación dentro de un rango de optimalidad aceptable y en tiempos de ejecución computacionales que hacen factible su implementación en la práctica operativa de las empresas. Con fundamento en los resultados presentados en el capítulo 9 derivamos las siguientes afirmaciones:
 - a. Podemos establecer a un 99% de confiabilidad que el AG propuesto alcanza un porcentaje de optimalidad > 90% en un tiempo computacional = 3 minutos.
 - b. Podemos establecer a un 93% de confiabilidad que el AG propuesto alcanza un porcentaje de optimalidad > 92.5% en un tiempo computacional = 5 minutos.
 - c. Podemos establecer a un 100% de confiabilidad que el AG propuesto alcanza un porcentaje de optimalidad > 92.5% en un tiempo computacional = 8 minutos.
5. Sin embargo, también debemos establecer que no obstante lo anterior, el algoritmo genético propuesto sólo puede asegurar un 90% de confiabilidad para alcanzar soluciones arriba del 95% de optimalidad y requiriendo tiempos computacionales de hasta 10 minutos.
6. Lo anterior nos conduce a establecer que la expectativa de solución del Algoritmo Genético (el propuesto y los otros dos que fueron comparados), se ve sensiblemente afectado en la medida en la cual se requieran soluciones que se acerquen al óptimo verdadero (exacto). Lo anterior aunque era previsible, hasta antes del experimento se desconocía cuantitativamente su efecto.

7. Así, se puede verificar en la tabla 9.2 en la columna ' $P(x=95\%)$ ', la forma en la cual va mermando drásticamente la confiabilidad del algoritmo genético propuesto al momento de ir disminuyendo la cantidad de tiempo computacional ofrecido para la solución. Como evidencia de lo anterior se tiene que:

Podemos establecer que el algoritmo genético propuesto sólo ofrece un 54% de confiabilidad cuando se requiere alcanzar un porcentaje de optimalidad = 95% en un tiempo computacional = 5 minutos. Lo anterior se traduce en el experimento al poderse alcanzar una optimalidad igual o mayor del 95% solo en 22 instancias de las 40 unidades experimentales que fueron aplicados para un límite de 5 minutos de tiempo computacional.

VI. En lo referente a la justificación económica de la propuesta algorítmica:

1. El análisis costo beneficio expuesto contribuyó a ofrecer las justificaciones económicas acerca de la viabilidad del algoritmo propuesto para mejorar la capacidad de transporte y el servicio al cliente sobre la base de supuestos conservadores definidos empíricamente por el investigador.
2. De acuerdo al análisis expuesto, la conveniencia económica para la aplicación del algoritmo genético está influenciado básicamente por tres factores:
 - d. Economía de escala de la empresa (# de Rutas en operación = EE)
 - e. Factor de Rutas a optimizar (optimización de la capacidad de transporte = FR)
 - f. Factor de Ventas a incrementar (optimización en el servicio al cliente = FV)
3. Aún en empresas debajo de 50 rutas (EE = 50), resulta recomendable la aplicación del algoritmo. Su contribución se verifica significativamente para valores de FR = 5%. Alternativamente también puede verificarse dicha contribución para valores de FV = 3% y de una manera mucho más importante cuando ambos factores se conjugan.
4. Del factor (EE), tenemos que para el 2do cuartil, es decir 50 = EE = 100, el máximo beneficio económico sería de \$4.9 Millones y el promedio de \$2.1 Millones por año.
5. En las empresas de gran escala (4to cuartil: 200 = EE = 500), tenemos que el máximo beneficio económico sería de \$26.7 Millones y el promedio de \$12.8 Millones por año.

VII. En lo referente al estudio analítico y predictivo de los parámetros del Algoritmo Genético:

1. El grado dimensional de los coeficientes obtenidos, contribuyen a establecer el efecto que cada uno de los parámetros individualmente ocasiona en el diseño del espacio solución.
2. Los resultados del experimento, apuntan a señalar los siguientes parámetros como significativos estadísticamente hablando: tamaño de la población (N), cantidad de generaciones (G) y factor de mutación (FM).
3. Con el diseño experimental desarrollado, se comprueba que el parámetro FC (factor de cruzamiento), no es estadísticamente significativo, lo cual confirma lo concluido por Pongcharoen ¹⁵⁶.
4. Los coeficientes de estimación para el modelo de regresión lineal desarrollado, evidencian que aumentar el tamaño de la población (N) favorece positivamente en la obtención de mejores soluciones en el algoritmo genético. Lo mismo puede decirse respecto de la cantidad de generaciones (G), aunque su impacto resulta menor en términos relativos respecto al anterior.
5. Para el coeficiente del factor de mutación (FM), también se verifica que aplicarlo con suficiente intensidad logra obtener mejores resultados en el algoritmo genético. No obstante los intervalos de confianza nos hace prever que éste parámetro debe ser cuidadosamente analizado sobre la base de un modelo de predicción cuadrático.
6. Para establecer conclusiones definitivas acerca de la definición de los niveles óptimos de operación para los parámetros del algoritmo genético propuesto, se recomienda desarrollar diseños experimentales del tipo “esparcido” tales como los propuestos por Stewardson y Hicks¹⁵⁷.

Con las conclusiones anteriores, el investigador establece un balance positivo de los objetivos y alcances definidos en el proyecto de investigación ¹⁵⁸.

¹⁵⁶ Pongcharoen, P. (2000), Using genetic algorithms for scheduling the production of capital goods, Proceedings of the 11th International working seminar on production economics, Austria, pag 441 -455.

¹⁵⁷ Supra: revisar pag 265.

¹⁵⁸ Supra: revisar los apartados 7.2 al 7.4

VIII. En lo referente a nuevas líneas de investigación sugeridas :

Se prevé que el desempeño obtenido por ambas propuestas de investigación, tanto la del algoritmo de ramificación y corte (BC) como la del algoritmo genético avanzado puedan servir para generar nuevas líneas de investigación particularmente para atender:

6. Instancias múltiples de problemas de logística de ruteo tales como el VRP-TW y el MPDP-TW.
7. Las instancias “no estáticas” de los problemas de ruteo, las cuales son frecuentes encontrar en las empresas que trabajan en ambientes de despacho dinámico.

En lo que respecta a ésta segunda línea de investigación, podemos afirmar que el tiempo de solución computacional se vuelve aún más crítico, ya que en estos casos no se dispone desde un principio del total de las órdenes a tener que atender a los clientes. Por tanto es requerido tomar en consideración el hecho de que las órdenes de entrega y/o de recolección de producto, van a irse incorporando al programa de ruteo a medida que va transcurriendo la jornada de trabajo lo cual implica lo siguiente:

1. El problema a resolver adquiere una mayor complejidad matemática.
2. Se requiere re-programar al estrategia de distribución más frecuentemente y en la misma medida en la que vayan ingresado en firme las órdenes de atención al cliente.
3. Se requiere un algoritmo que pueda ser ejecutado práctica y frecuentemente y que a la vez pueda ofrecer soluciones razonablemente aceptables (cerca al óptimo).

Tomando en cuenta los argumentos anteriores, el investigador ratifica la importancia que desde el punto de vista científico y práctico existe al promover una propuesta basada en la aplicación de un algoritmo genético. Los resultados expuestos en el presente proyecto de investigación, hacen prever que el algoritmo genético propuesto se vería aún más favorecido hacia su aplicación al ser empleado para atender problemas de ruteo en ambientes de despacho dinámico. Es comprensible que en dicho escenario el binomio “optimalidad” versus “tiempo computacional” se volverá aún más crítico.

i. Anexo A.

Fase de descomposición basada en el método SPP para el análisis y explotación topológica de la versión original de la red logística.

```
model Tspf2
  uses "mmxprs", "mmsystem"
  declarations
    kregen = 1      ! 1= Uno a la vez, 2 = Limite , 3 = Todos
    klim = 200     ! # de Nodos totales: Reales + Topológicos
    ktol = 0       !0.001
    kmaxtime = 100
    nodos = 1..klim
    kmotor = XPRS_BAR
    ki,kj,kw,kimay,krest,map,vrest,emap,ect,zcosto: integer
    wlim,ksec,dist,knmay,klen,ksol,emin,hsol: real
    zori,zdest: integer
    ktipo,kori,kdest: array(nodos) of integer
    costo: dynamic array(nodos,nodos) of real
    sol: dynamic array(nodos,nodos) of real
    x: dynamic array(nodos,nodos) of mpvar
    xrest: array(range) of lincpr
    cred: dynamic array(nodos,nodos) of real
  end-declarations

  ! VERSION ORIGINAL PARA EL MANEJO DEL TW SENCILLO
  ! INCLUYE 4 FASES DE DEPURACION
  ! XPRS_BAR, XPRS_PRI, XPRS_DUAL

  fopen("TOPO.TXT",F_INPUT)
  readln
  repeat
    readln(zori,zdest,zcosto)
    if zcosto >= costo(zori,zdest) then
      costo(zori,zdest) := zcosto
      create(x(zori,zdest))
      x(zori,zdest) is_binary
    end-if
  until (getparam('nbread') < 3)
  fclose(F_INPUT)

  writeln("PROCESO DE CARGA TERMINADO...")

  starttime := gettime
  finalize(nodos)

  setparam("XPRS_PRESOLVE",0)           ! SIN = 0, CON = 1
  setparam("XPRS_MIPRELSTOP",ktol)     ! TOLERANCIA
  setparam("XPRS_MAXTIME",kmaxtime)    ! # DE SEGUNDOS POR MIP
  !setparam("XPRS_CUTSTRATEGY",1)      ! 1=CONSERV, 2= AGRESIVO
  !setparam("XPRS_MIPPRESOLVE",2)     ! +1=Reduce Cost, +2=Logical Bin,
+4=Probing Bin
  !setparam("XPRS_MAXMIPSOL",20)       ! # DE SOLUCIONES ENTERAS
  !setparam("XPRS_MIPLOG",0)          ! SIN LOG EN MIP
```



```

!setparam("XPRS_CUTSTRATEGY",-1)      ! CUT AUTOMATICO

forall(i in nodos) do
  if i <= 100 then
    ktipo(i) := 1
  else
    ktipo(i) := 0
  end-if
end-do

fopen("RED.TXT",F_OUTPUT)
ki := 101
repeat
  kj := 101
  repeat
    if ki <> kj then
      writeln ("0", " ", " ", ki , " ", " ", kj, " ", " ", "0")
      forall(i in nodos) kori(i) := 0
      forall(i in nodos) kdest(i) := 0
      kori(ki) := 1
      kdest(kj) := 1

      forall(j in nodos)
        xrest(j) := kori(j) + sum(i in nodos) x(i,j) = sum(h in
nodos) x(j,h) + kdest(j)
      distancia := sum(i in nodos, j in nodos)
                    (costo(i,j)*x(i,j) + x(i,j))
      minimize(kmotor,distancia)
      if getprobstat = XPRS_INF then
        writeln ("error:", ki, " ", " ", kj)
        exit(ki)
      end-if

      hsol := sum(i in nodos, j in nodos) getsol(x(i,j))
      ksol := getobjval - hsol
      forall(i in nodos, j in nodos) sol(i,j) := getsol(x(i,j))

      zi := ki
      zj := 1
      cred(ki,kj) := ksol
      repeat
        if sol(zi,zj) = 1 then
          writeln("1" , " ", " ", zi," ", " ",zj, " ", " ", costo(zi,zj))
          sol(zi,zj) := 0
          zi := zj
          zj := 0
        end-if
        zj := zj + 1
      until zj > klim
      forall(j in nodos) xrest(j) -= xrest(j)
    end-if
    kj := kj + 1
  until kj > klim
  ki := ki + 1
until ki > klim
fclose(F_OUTPUT)

```

```
fopen("CRED.TXT",F_OUTPUT)
  forall(i in 101..klim, j in 101..klim)
    writeln(i, " ", j, " ", cred(i,j))
fclose(F_OUTPUT)

end-model
```

ii. Anexo B.

Fase de compresión vía estrategia de clusterización mediante criterios de maximización de afinidad.

```
model Tspf1
  uses "mmxprs", "mmsystem"
  declarations
    kred = 100
    klim = 50
    kbound = 15
    ktol = 0 !0.001
    kmaxtime = 20
    pto = 1..kred
    nodos = 1..klim
    knear = 1..kbound
    kmotor = XPRS_BAR
    ki,kj,kw,kimay,krest: integer
    wlim,ksec,dist,knmay,klen: real
    kx,ky,top,tcl: array(nodos) of real
    kdel,kpup: array(nodos) of integer
    kmaxclust = 3
    cdist: array(pto,pto) of real
    y: array(pto,pto) of mpvar
    z: array(pto) of mpvar
    zx,zy: array(pto) of real
    ztop,ztcl,zdel,zpup: array(pto) of integer
  end-declarations

  ! VERSION ORIGINAL PARA EL MANEJO DEL TW SENCILLO
  ! INCLUYE 4 FASES DE DEPURACION
  ! XPRS_BAR, XPRS_PRI, XPRS_DUAL

  fopen("TSPTW100.TXT",F_INPUT)
  kn:= 1
  repeat
  read(ki,zx(kn),zy(kn),ztop(kn),ztcl(kn),zdel(kn),zpup(kn))
  kn := kn + 1
  until (getparam('nbread') < 7)
  fclose(F_INPUT)
  starttime := gettime

  ! ***** INICIA FASE DE CLUSTERIZACION
  forall(i in pto, j in pto)
    cdist(i,j) := round(((zx(i) - zx(j))^2 +
      (zy(i) - zy(j))^2)^0.5)

  forall(i in pto, j in pto) y(i,j) is_binary
  forall(j in pto) z(j) is_binary

  forall(i in pto) sum(j in pto) y(i,j) = 1
  forall(j in pto) sum(i in pto) y(i,j) <= kmaxclust*z(j)
  sum(j in pto) z(j) = klim
  cluster:= sum(i in pto, j in pto) cdist(i,j)*y(i,j)
```

```

setparam("XPRS_PRESOLVE",0)           ! SIN = 0, CON = 1
setparam("XPRS_MIPPRESOLVE",0)       ! +1=Reduce Cost, +2=Logical Bin,
+4=Probing Bin
setparam("XPRS_CUTSTRATEGY",2)       ! CUT AUTOMATICO
setparam("XPRS_MAXTIME",kmaxtime)    ! # DE SEGUNDOS POR MIP

minimize(kmotor,cluster)

k := 0
forall(j in ptos) do
  wn := 0
  wx := 0.0
  wy := 0.0
  wop := klim
  wcl := 0
  wdel := 0
  wpup := 0
  forall(i in ptos) do
    if getsol(y(i,j)) >= 0.999 then
      wn := wn + 1
      wx := wx + zx(i)
      wy := wy + zy(i)
      wdel := wdel + zdel(i)
      wpup := wpup + zpup(i)
      if ztop(i) < wop then
        wop := ztop(i)
      end-if
      if ztcl(i) > wcl then
        wcl := ztcl(i)
      end-if
    end-if
  end-do
  if wn >= 1 then
    k := k + 1
    kx(k) := wx / wn
    ky(k) := wy / wn
    top(k) := wop
    tcl(k) := wcl
    kdel(k) := wdel
    kpup(k) := wpup
  end-if
end-do
writeln ("*** Termina Fase de Clusterización ***")
writeln ("Segundos de ejecución:", gettime-startime)
! ***** TERMINA FASE DE CLUSTERIZACION

fopen("TSPTW50.TXT",F_OUTPUT)
forall(i in nodos)
  writeln(i," ",kx(i)," ", ky(i)," ",round(top(i)/2),"
",round(tcl(i)/2)," ",kdel(i)," ",kpup(i))
fclose(F_OUTPUT)

fopen("CLUSTER.TXT",F_OUTPUT)
forall(i in ptos, j in ptos | getsol(y(i,j)) >= 0.999)
  writeln(i," ",j)

```

```
fclose(F_OUTPUT)
```

```
end-model
```

iii. Anexo C.

Fase de compresión discriminante de arcos basada en la heurística de los “k” nodos vecinos más cercanos para obtener la versión compacta de la red

```
model Tspf2
  uses "mmxprs", "mmsystem"
  declarations
    kregen = 3 ! 1= Uno a la vez, 2 = Limite , 3 = Todos
    klim = 50
    kbound = 15
    ktol = 0 !0.001
    kmaxtime = 100
    nodos = 1..klim
    knear = 1..kbound
    kmotor = XPRS_BAR
    ki,kj,kw,kimay,krest,map,vrest,emap,ect,kexit: integer
    wlim,ksec,dist,knmay,klen,ksol,emin,hsol: real
    knodo: array(knear) of integer
    kdist: array(knear) of real
    kx,ky,top,tcl: array(nodos) of real
    wdel,wpup:array(nodos) of integer
    twoc: dynamic array(nodos,nodos) of boolean
    twdif: dynamic array(nodos,nodos) of integer
    costo: dynamic array(nodos,nodos) of real
    msol,sol: dynamic array(nodos,nodos) of real
    rut3: dynamic array(nodos) of integer
    arco: dynamic array(nodos,nodos) of integer
    x: dynamic array(nodos,nodos) of mpvar
    zrest:array(range) of integer
    wrest,xrest: array(range) of linctr
    inicargo, wcargo: integer
    capcargo,wip: real
  end-declarations

  ! VERSION ORIGINAL PARA EL MANEJO DEL TW SENCILLO
  ! INCLUYE 4 FASES DE DEPURACION
  ! XPRS_BAR, XPRS_PRI, XPRS_DUAL

  fopen("TSPTW50.TXT",F_INPUT)
  kn:= 1
  repeat
    read(ki,kx(kn),ky(kn),top(kn),tcl(kn),wdel(kn),wpup(kn))
    kn := kn + 1
  until (getparam('nbread') < 7)
  fclose(F_INPUT)

  forall(i in nodos) do
    forall(j in nodos) do
      if j >= top(i) and j <= tcl(i) then
        twoc(i,j) := true
      else
        twoc(i,j) := false
      end-if
    end-do
  end-do
```

```

        end-do
    end-do

    kn := 1
    repeat
        ktw := 1
        repeat
            wscan := 0
            if not twoc(kn,ktw) then
                wscan := 1
                repeat
                    if ktw - wscan >= 1 then
                        xsc1 := ktw - wscan
                    else
                        xsc1 := klim + (ktw - wscan)
                    end-if
                    if ktw + wscan <= klim then
                        xsc2 := ktw + wscan
                    else
                        xsc2 := (ktw + wscan) - klim
                    end-if
                    if twoc(kn,xsc1) or twoc(kn,xsc2) then
                        break
                    end-if
                    wscan := wscan + 1
                until false
            end-if
            twdif(kn,ktw) := wscan
            ktw := ktw + 1
        until ktw > klim
        kn := kn + 1
    until kn > klim

    ki:= 1
    repeat
        kj:= 1
        forall (i in knear) kdist(i):= 1e10
        repeat
            dist := round(((kx(ki) - kx(kj))^2 +
                (ky(ki) - ky(kj))^2)^0.5)
            kk:= 1
            knmay:= 0
            kimay:= 0
            repeat
                if kdist(kk) > knmay then
                    if kdist(kk) = 1e10 then
                        kimay := kk
                        break
                    else
                        knmay := kdist(kk)
                        kimay := kk
                    end-if
                end-if
                kk:= kk + 1
            until kk > kbound
            if kimay <> 0 then

```

```

        kdist(kimay) := dist
        knodo(kimay) := kj
    end-if
    kj:= kj + 1
until kj > klim
kk:= 1
repeat
    if ki < knodo(kk) then
        costo(ki,knodo(kk)) := kdist(kk)
        create(x(ki,knodo(kk)))
        x(ki,knodo(kk)) is_binary
    else
        costo(knodo(kk),ki) := kdist(kk)
        create(x(knodo(kk),ki))
        x(knodo(kk),ki) is_binary
    end-if
    kk:= kk + 1
until kk > kbound
ki:= ki + 1
until ki > klim

writeln("PROCESO DE CARGA TERMINADO...")

```


iv. Anexo D.

Fase de generación agresiva de cortes aplicada a la versión compacta de la red

```
model Tspf2
  uses "mmxprs", "mmsystem"
  declarations
    kregen = 3 ! 1= Uno a la vez, 2 = Limite , 3 = Todos
    klim = 50
    kbound = 15
    ktol = 0 !0.001
    kmaxtime = 100
    nodos = 1..klim
    knear = 1..kbound
    kmotor = XPRS_BAR
  ki,kj,kw,kimay,krest,map,vrest,emap,ect,kexit: integer
  wlim,ksec,dist,knmay,klen,ksol,emin,hsol: real
  knodo: array(knear) of integer
  kdist: array(knear) of real
  kx,ky,top,tcl: array(nodos) of real
  wdel,wpup:array(nodos) of integer
  twoc: dynamic array(nodos,nodos) of boolean
  twdif: dynamic array(nodos,nodos) of integer
  costo: dynamic array(nodos,nodos) of real
  msol,sol: dynamic array(nodos,nodos) of real
  rut3: dynamic array(nodos) of integer
  arco: dynamic array(nodos,nodos) of integer
  x: dynamic array(nodos,nodos) of mpvar
  zrest:array(range) of integer
  wrest,xrest: array(range) of lincpr
  inicargo, wcargo: integer
  capcarga,wip: real
end-declarations

! VERSION ORIGINAL PARA EL MANEJO DEL TW SENCILLO
! INCLUYE 4 FASES DE DEPURACION
! XPRS_BAR, XPRS_PRI, XPRS_DUAL

fopen("TSPTW50.TXT",F_INPUT)
kn:= 1
repeat
  read(ki,kx(kn),ky(kn),top(kn),tcl(kn),wdel(kn),wpup(kn))
  kn := kn + 1
until (getparam('nbread') < 7)
fclose(F_INPUT)

forall(i in nodos) do
  forall(j in nodos) do
    if j >= top(i) and j <= tcl(i) then
      twoc(i,j) := true
    else
      twoc(i,j) := false
    end-if
  end-do
end-do
```

```

writeln("PROCESO DE CARGA TERMINADO...")

starttime := gettime
finalize(nodos)

forall(h in nodos)
    sum(i in 1..h-1) x(i,h) + sum(j in h+1..klim) x(h,j) = 2

distancia:= sum(i in nodos, j in i+1..klim)
                                costo(i,j)*x(i,j)

setparam("XPRS_PRESOLVE",0)          ! SIN = 0, CON = 1
setparam("XPRS_MIPRELSTOP",kto1)     ! TOLERANCIA
setparam("XPRS_MAXTIME",kmaxtime)    ! # DE SEGUNDOS POR MIP
!setparam("XPRS_CUTSTRATEGY",1)      ! 1=CONSERV, 2= AGRESIVO
!setparam("XPRS_MIPPRESOLVE",2)      ! +1=Reduce Cost, +2=Logical Bin,
+4=Probing Bin
!setparam("XPRS_MAXMIPSOL",20)       ! # DE SOLUCIONES ENTERAS
setparam("XPRS_MIPLOG",0)            ! SIN LOG EN MIP
!setparam("XPRS_CUTSTRATEGY",-1)     ! CUT AUTOMATICO

procedure bucout
! ***** IMPRIME EL ARCHIVO DEL TOUR *****
forall(i in nodos, j in nodos) sol(i,j) := msol(i,j)
fopen("TOUR.TXT",F_OUTPUT)
ki := emap
kj := 1
ktw := 0
repeat
    if ki<kj then
        if sol(ki,kj) = 1 then
            ktw := ktw + 1
            writeln(ki," ",kj," ",ktw," ",top(kj)," ",tcl(kj))
            sol(ki,kj) := 0
            ki := kj
            kj := 0
        end-if
    else
        if sol(kj,ki) = 1 then
            ktw := ktw + 1
            writeln(ki," ",kj," ",ktw," ",top(kj)," ",tcl(kj))
            sol(kj,ki) := 0
            ki := kj
            kj := 0
        end-if
    end-if
    kj := kj + 1
until kj > klim
fclose(F_OUTPUT)
end-procedure

```

```

procedure bucmenor
  if ksol <= emin then
    ect := 1
    emin := ksol
    emap := map
    forall(i in nodos, j in nodos) msol(i,j) := getsol(x(i,j))
    bucout
  else
    if ksol = emin then
      ect := ect + 1
    end-if
  end-if
end-procedure

```

```

procedure bucborra
  zc := 1
  repeat
    if zrest(zc) = 1 then
      wrest(zc) -= wrest(zc)
      zrest(zc) := 0
      vrest := vrest - 1
      if kregen = 1 then
        break
      else
        if kregen = 2 then
          if vrest <= 20 then
            break
          end-if
        end-if
      end-if
    end-if
    zc := zc + 1
  until zc > krest
end-procedure

```

```

procedure buc1
  kc := 1
  repeat
    minimize(kmotor,distancia)
    if getprobat = XPRS_INF then
      break
    end-if
    ksol := getobjval
    savebasis(1)

    forall(i in nodos, j in nodos) sol(i,j) := getsol(x(i,j))

    kmay := 0
    kmen := klim
    kr := 0
    repeat
      forall(i in nodos) rut3(i) := 0

      ki := 1

```

```

kp1:=0
repeat
  kj:= 1
  repeat
    if sol(ki,kj) = 1 then
      kp1:=ki
      break 2
    end-if
    kj := kj + 1
  until kj > klim
  ki := ki + 1
until ki > klim

if kp1 = 0 then
  break      ! se acabaron los sub-tours
end-if

kp1 := 1
rut3(ki) := 1
kj := 1
kr := kr + 1
repeat
  if ki<kj then
    if sol(ki,kj) = 1 then
      kp1 := kp1 + 1
      rut3(kj) := 1
      sol(ki,kj) := 0
      ki := kj
      kj := 0
    end-if
  else
    if sol(kj,ki) = 1 then
      kp1 := kp1 + 1
      rut3(kj) := 1
      sol(kj,ki) := 0
      ki := kj
      kj := 0
    end-if
  end-if
  kj := kj + 1
until kj > klim
kp1 := kp1 - 1

if kp1 <= kmen and kp1 >= 2 then
  kmen := kp1
end-if
if kp1 >= kmay then
  kmay:= kp1
end-if

if kp1 = klim then
  kz := 1      ! entra al time window
  mft := klim
  repeat
    forall(i in nodos, j in nodos) sol(i,j):=

```

getsol(x(i,j))

```

ki := kz
kj := 1
wok := 1
ktw := 1
ft := 0
klen := 0
repeat
  if ki < kj then
    if sol(ki,kj) = 1 then
      if not twoc(kj,ktw) then
        wok := 0
        ft := ft + 1 ! fuera de Time

        kdif := twdif(kj,ktw)
        if kdif > klen then
          wj := kj
          wtw := ktw
          klen := kdif
        end-if
      end-if
      ktw := ktw + 1
      sol(ki,kj) := 0
      ki := kj
      kj := 0
    end-if
  else
    if sol(kj,ki) = 1 then
      if not twoc(kj,ktw) then
        wok := 0
        ft := ft + 1 ! fuera de Time

        kdif := twdif(kj,ktw)
        if kdif > klen then
          wj := kj
          wtw := ktw
          klen := kdif
        end-if
      end-if
      ktw := ktw + 1
      sol(kj,ki) := 0
      ki := kj
      kj := 0
    end-if
  end-if
  kj := kj + 1
until kj > klim
if ft < mft then
  mft := ft          ! menor cantidad de
  map := kz          ! Nodo de inicio
  kap := wj          ! Nodo con la mayor violación
  ztw := wtw         ! Hr de Llegada del Nodo con
  mayor violación de TW
end-if

```

Windows en kj

Windows en kj

violaciones al TW

de TW

mayor violación de TW

```

        if wok = 1 then
            kc := kc + 1
            writeln (ksol, " = ", kmen, ",", kmay, ",", kr,
" Ti= ", gettime-starttime)
            break 3
        end-if
        kz := kz + 1
    until kz > klim
    ! A CONTINUACION SE AGREGA EL CUT DE LA RUTA GLOBAL
    forall(i in nodos, j in nodos) sol(i,j) := getsol(x(i,j))
    sum(k1 in nodos, k2 in nodos |
        sol(k1,k2) = 1 and k1<k2 )
        x(k1,k2) <= kp1-1

    ! A CONTINUACION SE IMPRIME LA MEJOR SECUENCIA DEL TOUR
    forall(i in nodos, j in nodos) arco(i,j) := 0
    forall(i in nodos, j in nodos) sol(i,j) := getsol(x(i,j))
    ki := map
    kj := 1
    ktw := 1
    repeat
        if ki<kj then
            if sol(ki,kj) = 1 then

                if ktw<>1 and ktw<>klim then
                    if twoc(kap,ktw+1) and

                        if kap < kj then
                            arco(kap,kj) := 1
                        else
                            arco(kj,kap) := 1
                        end-if
                    end-if

                end-if
                sol(ki,kj) := 0
                ktw := ktw + 1
                ki := kj
                kj := 0
            end-if
        else
            if sol(kj,ki) = 1 then
                if ktw<>1 and ktw<>klim then
                    if twoc(kap,ktw+1) and

                        if kap < kj then
                            arco(kap,kj) := 1
                        else
                            arco(kj,kap) := 1
                        end-if
                    end-if

                end-if
                sol(kj,ki) := 0
                ktw := ktw + 1
                ki := kj
                kj := 0
            end-if
        end-if
    twoc(kap,ktw-1) then
    twoc(kap,ktw-1) then

```

```

        end-if
        kj := kj + 1
until kj > klim

! A CONTINUACION SE AGREGA EL SUB-CUT PARCIAL
krest := krest + 1
zrest(krest) := 1
wrest(krest) := sum(k1 in nodos, k2 in nodos |
                    arco(k1,k2) = 1 )
                    x(k1,k2) >= 1
vrest := vrest + 1
writeln ("sale del TW con: ", mft, " Rest= ", vrest)
writeln(kap,"","zrw","", top(kap),"",tc1(kap))

end-if

if kp1 < klim then
! SE AGREGA EL SUB-TOUR ELIMINATION CONSTR
sum(k1 in nodos, k2 in nodos |
    rut3(k1) = 1 and rut3(k2) = 1 and k1<k2 )
    x(k1,k2) <= kp1-1
end-if
until kp1 <= 0 or kp1 >= klim

loadprob(distancia)
loadbasis(1)

kc := kc + 1
writeln (ksol, " = ", kmen, ",", kmay, ",", kr, " Ti= ", gettime-
starttime)
if ksol > 1.05*hsol then
    bucborra
end-if
until false
hsol := ksol
end-procedure

!*****
!***** PROCEDIMIENTO PARA LA CONSTRUCCION DE LOS CUTS PARA EL PDP *****
!*****

procedure bucpdp
    kc :=1
    repeat
        minimize(kmotor,distancia)
        if getprobstat = XPRS_INF then
            writeln ("*** E R R O R ***")
            break
        end-if
        ksol := getobjval
        savebasis(1)
        forall(i in nodos, j in nodos) sol(i,j) := getsol(x(i,j))

        kmay := 0
        kmen := klim
        kr := 0

```

```

repeat
  forall(i in nodos) rut3(i) := 0
  ki:= 1
  kp1:=0
  repeat
    kj:= 1
    repeat
      if sol(ki,kj) = 1 then
        kp1:=ki
        break 2
      end-if
      kj := kj + 1
    until kj > klim
    ki := ki + 1
  until ki > klim

  if kp1 = 0 then
    break      ! se acabaron los sub-tours
  end-if

  kp1 := 1
  rut3(ki) := 1
  kj := 1
  kr := kr + 1
  repeat
    if ki<kj then
      if sol(ki,kj) = 1 then
        kp1 := kp1 + 1
        rut3(kj):= 1
        sol(ki,kj) := 0
        ki := kj
        kj := 0
      end-if
    else
      if sol(kj,ki) = 1 then
        kp1 := kp1 + 1
        rut3(kj):= 1
        sol(kj,ki) := 0
        ki := kj
        kj := 0
      end-if
    end-if
    kj := kj + 1
  until kj > klim
  kp1 := kp1 - 1

  if kp1 <= kmen and kp1 >= 2 then
    kmen := kp1
  end-if
  if kp1 >= kmay then
    kmay:= kp1
  end-if

  if kp1 = klim then
    kz := 1      ! Entra al procedimiento de cuts para el PDP
    mft := klim

```



```

repeat
  forall(i in nodos, j in nodos) sol(i,j):=
    ki := kz
    kj := 1
    wok := 1
    ktw := inicargo
    ft := 0
    klen := 0
    repeat
      if ki<kj then
        if sol(ki,kj) = 1 then
          ktw := ktw - wdel(kj) + wpup(kj)
          if ktw > capcarga then
            wok := 0
            ft := ft + 1 ! Excede

            kdif := ktw - capcarga
            if kdif > klen then
              wj := kj
              wtw := ktw
              klen := kdif
            end-if
          end-if
          sol(ki,kj) := 0
          ki := kj
          kj := 0
        end-if
      else
        if sol(kj,ki) = 1 then
          ktw := ktw - wdel(kj) + wpup(kj)
          if ktw > capcarga then
            wok := 0
            ft := ft + 1 ! Excede

            kdif := ktw - capcarga
            if kdif > klen then
              wj := kj
              wtw := ktw
              klen := kdif
            end-if
          end-if
          sol(kj,ki) := 0
          ki := kj
          kj := 0
        end-if
      end-if
      kj := kj + 1
    until kj > klim
    if ft < mft then
      mft := ft ! menor cantidad de violaciones al
      map := kz ! Apuntador al Nodo de inicio
      kap := wj ! Nodo con la mayor violación del
  end-repeat

```

getsol(x(i,j))

capacidad de carga en kj

capacidad de carga en kj

PDP

PDP

```

                                ztw := wtw ! Carga del Nodo con mayor
violación al PDP
                                end-if
                                if wok = 1 then
                                    map := kz
                                    kc := kc + 1
                                    writeln (ksol, " = ", kmen, ",", kmay, ",", kr,
" Ti= ", gettime-starttime)
                                break 3
                                end-if
                                kz := kz + 1
                                until kz > klim
                                ! A CONTINUACION SE AGREGA EL CUT DE LA RUTA GLOBAL
                                forall(i in nodos, j in nodos) sol(i,j) := getsol(x(i,j))
                                sum(k1 in nodos, k2 in nodos |
                                    sol(k1,k2) = 1 and k1<k2 )
                                    x(k1,k2) <= kp1-1

                                ! SE BUSCAN LAS SEC'S HACIA DONDE IR A UBICAR EL EXCESO DE
CARGA
                                forall(i in nodos, j in nodos) arco(i,j) := 0
                                forall(i in nodos, j in nodos) sol(i,j) := getsol(x(i,j))
                                ki := map
                                kj := 1
                                ktw := inicargo
                                repeat
                                    if ki<kj then
                                        if sol(ki,kj) = 1 then
                                            ktw := ktw - wdel(kj) + wpup(kj)
                                            if ktw - wdel(kap) + wpup(kap) < capcarga
then
                                                if kap < kj then
                                                    arco(kap,kj) := 1
                                                else
                                                    arco(kj,kap) := 1
                                                end-if
                                            end-if
                                            sol(ki,kj) := 0
                                            ki := kj
                                            kj := 0
                                        end-if
                                    else
                                        if sol(kj,ki) = 1 then
                                            ktw := ktw - wdel(kj) + wpup(kj)
                                            if ktw - wdel(kap) + wpup(kap) < capcarga
then
                                                if kap < kj then
                                                    arco(kap,kj) := 1
                                                else
                                                    arco(kj,kap) := 1
                                                end-if
                                            end-if
                                            sol(kj,ki) := 0
                                            ki := kj
                                            kj := 0
                                        end-if
                                end-if

```

```

                end-if
                kj := kj + 1
            until kj > klim

                ! A CONTINUACION SE AGREGA EL SUB-CUT PARCIAL
                krest := krest + 1
                zrest(krest) := 1
                wrest(krest) := sum(k1 in nodos, k2 in nodos |
                                    arco(k1,k2) = 1 )
                                    x(k1,k2) >= 1

                vrest := vrest + 1
                writeln ("sale del PDP con: ", mft, " Rest= ", vrest)
                writeln(kap,"","zrw","", capcarga)
            end-if

            if kp1 < klim then
                ! SE AGREGA EL SUB-TOUR ELIMINATION CONSTR
                sum(k1 in nodos, k2 in nodos |
                    rut3(k1) = 1 and rut3(k2) = 1 and k1<k2 )
                    x(k1,k2) <= kp1-1
            end-if
            until kp1 <= 0 or kp1 >= klim

            loadprob(distancia)
            loadbasis(1)

            kc := kc + 1
            writeln (ksol, " = ", kmen, ",", kmay, ",", kr, " Ti= ", gettime-
starttime)
            until false
            hsol := ksol
            if getprobstat <> XPRS_INF then
                bucmenor
            end-if
        end-procedure

procedure buc2
    ! ***** INICIA DEPURACION *****
    if krest > 0 and vrest > 0 then
        writeln ("*** Depuración:", ksol, " = ", vrest, " Best= ",emin)
        forall(i in 1..krest) xrest(i) := wrest(i)

        kc:= 1
        repeat
            if zrest(kc) = 1 then
                wrest(kc) -= wrest(kc)
                loadprob(distancia)
                loadbasis(1)
                minimize(kmotor,distancia)
                ksol := getobjval
                savebasis(1)
                kz := 1 ! entra al Time Window y al PDP
                repeat
                    forall(i in nodos, j in nodos) sol(i,j):= getsol(x(i,j))

```

```

ki := kz
kj := 1
wok := 1
ktw := 0
wcargo := inicargo
repeat
  if ki < kj then
    if sol(ki,kj) = 1 then
      ktw := ktw + 1
      wcargo := wcargo - wdel(kj) + wpup(kj)
      if not twoc(kj,ktw) or wcargo > capcarga
then
        wok := 0
        break
      end-if
      sol(ki,kj) := 0
      ki := kj
      kj := 0
    end-if
  else
    if sol(kj,ki) = 1 then
      ktw := ktw + 1
      wcargo := wcargo - wdel(kj) + wpup(kj)
      if not twoc(kj,ktw) or wcargo > capcarga
then
        wok := 0
        break
      end-if
      sol(kj,ki) := 0
      ki := kj
      kj := 0
    end-if
  end-if
  kj := kj + 1
until kj > klim
if wok = 1 then
  map := kz
  break
end-if
kz := kz + 1
until kz > klim
if wok = 1 and ktw >= klim then
  vrest := vrest - 1
  writeln("F1 Borra:", kc, " = ", ksol, " Rest= ", vrest, "
Best= ",emin)
  zrest(kc) := 0
  bucmenor
else
  wrest(kc) += xrest(kc)
end-if
end-if
kc := kc + 1
until kc > krest

! ***** FASE INTERMEDIA DE DEPURACION *****
repeat

```

```

zsig := 0
kc:= 1
repeat
  if zrest(kc) = 1 then
    ztime := gettime
    wrest(kc) -= wrest(kc)
    loadprob(distancia)
    loadbasis(1)
    minimize(kmotor,distancia)
    ksol := getobjval
    savebasis(1)
    kcic := 0
    repeat
      forall(i in nodos, j in nodos) sol(i,j) := getsol(x(i,j))
      kmay := 0
      kmen := klim
      kr := 0
      repeat
        forall(i in nodos) rut3(i) := 0
        ki:= 1
        kp1:=0
        repeat
          kj:= 1
          repeat
            if sol(ki,kj) = 1 then
              kp1:=ki
              break 2
            end-if
            kj := kj + 1
          until kj > klim
          ki := ki + 1
        until ki > klim
      until kp1 = 0 then
        break      ! se acabaron los sub-tours
      end-if

      kp1 := 1
      rut3(ki) := 1
      kj := 1
      kr := kr + 1
      repeat
        if ki<kj then
          if sol(ki,kj) = 1 then
            kp1 := kp1 + 1
            rut3(kj) := 1
            sol(ki,kj) := 0
            ki := kj
            kj := 0
          end-if
        else
          if sol(kj,ki) = 1 then
            kp1 := kp1 + 1
            rut3(kj) := 1
            sol(kj,ki) := 0
            ki := kj
          end-if
        end-if
      until kp1 = 0 then
        break
      end-if
    until kcic = 0
  end-if
  kc := kc + 1
until zsig = 1

```

```

                                kj := 0
                            end-if
                        end-if
                    kj := kj + 1
                until kj > klim
                kp1 := kp1 - 1

            if kp1 = klim then
                kz := 1      ! entra al time window y al PDP
                repeat
                    forall(i in nodos, j in nodos) sol(i,j):=
getsol(x(i,j))

                                ki := kz
                                kj := 1
                                wok := 1
                                ktw := 0
                                wcargo := inicargo
                                repeat
                                    if ki < kj then
                                        if sol(ki,kj) = 1 then
                                            ktw := ktw + 1
                                            wcargo := wcargo -
wdel(kj) + wpup(kj)
                                            if not twoc(kj,ktw) or
wcargo > capcarga then
                                                wok := 0
                                                break
                                            end-if
                                            sol(ki,kj) := 0
                                            ki := kj
                                            kj := 0
                                        end-if
                                    else
                                        if sol(kj,ki) = 1 then
                                            ktw := ktw + 1
                                            wcargo := wcargo -
wdel(kj) + wpup(kj)
                                            if not twoc(kj,ktw) or
wcargo > capcarga then
                                                wok := 0
                                                break
                                            end-if
                                        sol(kj,ki) := 0
                                        ki := kj
                                        kj := 0
                                        end-if
                                    end-if
                                end-if
                            kj := kj + 1
                        until kj > klim
                        if wok = 1 then
                            map := kz
                            break 3
                        end-if
                    kz := kz + 1
                until kz > klim

```

```

! SE AGREGA EL CUT DE LA RUTA GLOBAL
forall(i in nodos, j in nodos) sol(i,j):=
getsol(x(i,j))
sum(k1 in nodos, k2 in nodos |
sol(k1,k2) = 1 and k1<k2 )
x(k1,k2) <= kp1-1
end-if
if kp1 < klim then
! SE AGREGA EL SUB-TOUR ELIMINATION CONSTR
sum(k1 in nodos, k2 in nodos |
rut3(k1) = 1 and rut3(k2) = 1 and
k1<k2 )
x(k1,k2) <= kp1-1
end-if
until kp1 <= 0 or kp1 >= klim
loadprob(distancia)
loadbasis(1)
minimize(kmotor,distancia)
ksol := getobjval
savebasis(1)
kcic := kcic + 1
until kp1 >= klim
if wok = 0 then
wrest(kc) += xrest(kc)
else
vrest := vrest - 1
writeln("F2 Borra:", kc, " = ", ksol, " Rest= ", vrest, "
Best= ", emin)
zrest(kc) := 0
zsig := 1
bucmenor
end-if
end-if
kc := kc + 1
until kc > krest
! ***** TERMINA FASE INTERMEDIA DE DEPURACION *****
until zsig = 0 or vrest <= 0
end-if
end-procedure

! ***** PROCEDIMIENTO PRINCIPAL *****
emin := 1e6
ect := 0
krest := 0
vrest := 0
zfirst := 0
hsol := 1e6
kexit := 0
inicargo := sum(i in nodos) wdel(i)
capcarga := maxlist(sum(i in nodos) wdel(i), sum(i in nodos) wpup(i))*1.9
writeln ("DEL:",sum(i in nodos) wdel(i), " PUP:",sum(i in nodos) wpup(i), "
CAP:",capcarga)

repeat

```

```

buc1
bucpdp
if getprobstat = XPRS_INF then
    bucborra
else
    buc2
    if vrest = 0 then
        kexit := 1
    else
        bucborra
    end-if
end-if
until gettime-starttime > 100000 or kexit = 1

! ***** FINALIZA EJECUCION DEL MODELO *****
ksol := emin
forall(i in nodos, j in nodos) sol(i,j) := msol(i,j)
writeln ("*** Termina:", ksol)

ki := emap
kj := 1
ktw := 0
wcargo := inicargo
wip := 0
repeat
    if ki < kj then
        if sol(ki,kj) = 1 then
            ktw := ktw + 1
            wcargo := wcargo - wdel(kj) + wpup(kj)
            wip := wip + wcargo
            writeln(ki," ",kj," ",ktw," ",top(kj)," ",tcl(kj)," W=
",wdel(kj), " ", wpup(kj), " ", wcargo)
            sol(ki,kj) := 0
            ki := kj
            kj := 0
        end-if
    else
        if sol(kj,ki) = 1 then
            ktw := ktw + 1
            wcargo := wcargo - wdel(kj) + wpup(kj)
            wip := wip + wcargo
            writeln(ki," ",kj," ",ktw," ",top(kj)," ",tcl(kj)," W=
",wdel(kj), " ", wpup(kj), " ", wcargo)
            sol(kj,ki) := 0
            ki := kj
            kj := 0
        end-if
    end-if
    kj := kj + 1
until kj > klim
writeln("Tiempo: ",gettime-starttime)
wip := wip / 50
writeln("WIP: ",wip)

end-model

```


v. Anexo E.

Fase evolutiva generacional para explotar el "pool" de cortes generados.

```
Sub mipexa()  
  
Dim i, j, k, n As Integer  
Dim kmen, kref As Double  
Dim karcos, kxy, kagrupa, kinforef, kapert, kcierre As Variant  
  
SolverOk SetCell:="$B$198", MaxMinVal:=2, ValueOf:=0, ByChange:="Arcos,Clusters",  
Engine _  
:=4, EngineDesc:="XPRESS LP/MIP Solver"  
SolverSolve (True)  
  
kxy = Application.Names("theLocations").RefersTo  
karcos = Application.Names("arcos").RefersTo  
kagrupa = Application.Names("agrupa").RefersTo  
kinforef = Application.Names("inforef").RefersTo  
kapert = Application.Names("apertura").RefersTo  
kcierre = Application.Names("cierre").RefersTo  
  
Range(kinforef).Select  
Selection.ClearContents  
  
n = Range(karcos).Rows.Count  
kref = 0  
  
k = 0  
For j = 1 To n  
    h = 0  
    x = 0  
    y = 0  
    kent = 1000#  
    ksal = 0  
    For i = 1 To n  
        If Range(karcos).Rows(i).Columns(j).Value >= 0.999 Then  
            If h = 0 Then  
                k = k + 1  
                h = h + 1  
                Range(kinforef).Rows(k).Columns(h).Value = k  
            End If  
            h = h + 1  
            Range(kinforef).Rows(k).Columns(h).Value = i  
            x = x + Range(kxy).Rows(i).Columns(1)  
            y = y + Range(kxy).Rows(i).Columns(2)  
            If Range(kapert).Rows(i).Value < kent Then  
                kent = Range(kapert).Rows(i).Value  
            End If  
            If Range(kcierre).Rows(i).Value > ksal Then  
                ksal = Range(kcierre).Rows(i).Value  
            End If  
        End If  
    Next i  
    If x <> 0 Then
```

```

        Range(kinforef).Rows(k).Columns(5).Value = x / (h - 1)
        Range(kinforef).Rows(k).Columns(6).Value = y / (h - 1)
        Range(kinforef).Rows(k).Columns(7).Value = kent
        Range(kinforef).Rows(k).Columns(8).Value = ksal
        Range(kinforef).Rows(k).Columns(9).Value = h - 1
    End If
Next j

```

```

'*****

```

```

'***** SE EJECUTA EL PROCEDIMIENTNO DE MIRUTA

```

```

Sub miruta()

```

```

Dim i, j, k, kord, n, kp, kt, kp1, kp2, kpasso, ki As Integer
Dim ktrial, kmen, kdist1, kdist2 As Double
Dim RangeOrder As Variant
Dim ksec(100), ktour(100), ktourx(100), ktoury(100) As Integer
Dim ksigue As Boolean

```

```

RangeOrder = Application.Names("theOrder").RefersTo
n = Range(RangeOrder).Rows.Count

```

```

ksigue = True
i = 1
kord = 1
Range(RangeOrder).Rows(kord) = i
ktour(kord) = i
ksec(i) = 9999

```

```

While ksigue
    kmen = 999999
    ksigue = False
    For kp = 1 To kord
        For j = 1 To n
            If ksec(j) <> 9999 Then
                ktrial = midistancia(ktour(kp), j)
                If ktrial < kmen Then
                    kmen = ktrial
                    kt = kp
                    k = j
                    ksigue = True
                End If
            End If
        Next j
    Next kp
    If ksigue = True Then
        kord = kord + 1
        Range(RangeOrder).Rows(kord) = k
        ktour(kord) = k
        ksec(k) = 9999
        i = k
    End If
    If kord >= 4 Then
        kp1 = kt - 1
        kp2 = kt + 1
        If kt = 1 Then

```

```

    kp1 = kord - 1
End If
If kt = kord - 1 Then
    kp2 = 1
End If

```

```

If kt = 1 Then
    For ki = 1 To kord
        ktourx(ki) = ktour(ki)
    Next ki
End If
If kt = kord - 1 Then
    For ki = 1 To kp1
        ktourx(ki) = ktour(ki)
    Next ki
    ktourx(kord - 1) = ktour(kord)
    ktourx(kord) = ktour(kord - 1)
End If

```

```

If kt <> 1 And kt <> kord - 1 Then
    For ki = 1 To kp1
        ktourx(ki) = ktour(ki)
    Next ki
    ktourx(kt) = ktour(kord)
    For ki = kp2 To kord
        ktourx(ki) = ktour(ki - 1)
    Next ki
End If

```

```

If kt = kord - 1 Then
    For ki = 1 To kord
        ktoury(ki) = ktour(ki)
    Next ki
End If
If kt = 1 Then
    ktoury(1) = ktour(1)
    ktoury(2) = ktour(kord)
    For ki = 3 To kord
        ktoury(ki) = ktour(ki - 1)
    Next ki
End If
If kt <> 1 And kt <> kord - 1 Then
    For ki = 1 To kt
        ktoury(ki) = ktour(ki)
    Next ki
    ktoury(kp2) = ktour(kord)
    For ki = kp2 + 1 To kord
        ktoury(ki) = ktour(ki - 1)
    Next ki
End If

```

```

kdist1 = 0
For ki = 1 To kord - 1
    kdist1 = kdist1 + midistancia(ktourx(ki), ktourx(ki + 1))
Next ki

```

```

kdist1 = kdist1 + midistancia(ktourx(kord), ktourx(1))

kdist2 = 0
For ki = 1 To kord - 1
    kdist2 = kdist2 + midistancia(ktoury(ki), ktoury(ki + 1))
Next ki
kdist2 = kdist2 + midistancia(ktoury(kord), ktoury(1))

If kdist1 <= kdist2 Then
    For ki = 1 To kord
        ktour(ki) = ktourx(ki)
        Range(RangeOrder).Rows(ki) = ktour(ki)
    Next ki
Else
    For ki = 1 To kord
        ktour(ki) = ktoury(ki)
        Range(RangeOrder).Rows(ki) = ktour(ki)
    Next ki
End If
End If
Wend
'*****

k2apert = Application.Names("ajapert").RefersTo
k2cierre = Application.Names("ajcierre").RefersTo

n2 = Range(k2apert).Rows.Count

For m = 1 To n2
    For i = 1 To k
        For j = 1 To 3
            If Range(kinforef).Rows(i).Columns(j + 1).Value = m Then
                Range(k2apert).Rows(m).Value =
                    Range(kinforef).Rows(i).Columns(7).Value
                Range(k2cierre).Rows(m).Value =
                    Range(kinforef).Rows(i).Columns(8).Value
                j = 100
                i = 100
            End If
        Next j
    Next i
Next m

'*****
'***** SE EJECUTA EL PROCEDIMIENTNO DE SECFIN2
secfin2
'*****

Sub secfin2()

Dim i, j, k, nord, nsec, ki As Integer
Dim p(3) As Integer
Dim kv1(100) As Integer
Dim kmen, kref As Double
Dim kvar(100) As String

```

```

Dim korder, ksecfin, kagrupa, ksecref As Variant
Dim kcom1 As String

Sheets("INP").Select

korder = Application.Names("theorder").RefersTo
kagrupa = Application.Names("agrupa").RefersTo
ksecfin = Application.Names("secfin").RefersTo

Range(ksecfin).Select
Selection.ClearContents

nord = Range(korder).Rows.Count
nsec = Range(ksecfin).Rows.Count

kcolj = 1
kcolv = 6
kv = 1
For i = 1 To nord
    ki = Range(korder).Rows(i).Value
    p(1) = Range(kagrupa).Rows(ki).Columns(1).Value
    p(2) = Range(kagrupa).Rows(ki).Columns(2).Value
    p(3) = Range(kagrupa).Rows(ki).Columns(3).Value
    klen = Sgn(p(1)) + Sgn(p(2)) + Sgn(p(3))
    If klen = 1 Then
        Range(ksecfin).Rows(kcolj).Value = p(1)
        kcolj = kcolj + 1
    Else
        kini = kcolv
        kvar(kv) = "U" & kini & ":U" & kini + klen - 1
        kv1(kv) = klen
        kv = kv + 1
        For k = 1 To klen
            kcom2 = "=V" & kcolv
            Range(ksecfin).Rows(kcolj).Formula = kcom2
            kcom1 = "U" & kcolv
            Range(kcom1).Value = k
            kcom1 = "V" & kcolv
            kcom2 = "=INDEX(W" & kini & ":W" & kini + klen - 1 & ",U" & kcolv &
",1)"
            Range(kcom1).Formula = kcom2
            kcom1 = "W" & kcolv
            Range(kcom1).Value = p(k)
            kcolj = kcolj + 1
            kcolv = kcolv + 1
        Next k
    End If
Next i

SolverReset

kcom1 = "U6:U" & kcolv - 1
SolverAdd cellRef:=kcom1, relation:=3, formulaText:="$BA$1"

For i = 1 To kv - 1
    If kv1(i) = 2 Then

```

```

        kcom1 = "$BA$2"
    Else
        kcom1 = "$BA$3"
    End If
    SolverAdd cellRef:=kvar(i), relation:=1, formulaText:=kcom1
    SolverAdd cellRef:=kvar(i), relation:=4
    SolverAdd cellRef:=kvar(i), relation:=6
Next i

SolverEVOptions MaxTime:=10000, Iterations:=100000, Precision:=0.000001, _
    Convergence:=0.0001, PopulationSize:=0, MutationRate:=0.075,
RequireBounds:= _
    True, StepThru:=False, Scaling:=False, AssumeNonneg:=True, BypassReports:=
-
    False, LocalSearch:=4
SolverLimOptions MaxSubProblems:=500000, MaxFeasibleSols:=500000, Tolerance:=0.05,
-
    MaxTimeNoImp:=100, SolveWithout:=False

kcom1 = "U6:U" & kcolv - 1
SolverOk SetCell:="$R$16", MaxMinVal:=2, ValueOf:=0, ByChange:=kcom1, Engine _
    :=3, EngineDesc:="Standard Evolutionary"

SolverSolve (True)
End Sub

```

vi. Anexo F.

Fase de descompresión de la ruta propuesta para la versión original de la red

```
model Tspf3
  uses "mmxprs", "mmsystem"
  declarations
    kred = 100
    klim = 50
    kbound = 10
    ktol = 0 !0.001
    kmaxtime = 100
    ptos = 1..kred
    nodos = 1..klim
    knear = 1..kbound
    kmotor = XPRS_BAR
  ki,kj,kw,kimay,krest: integer
  wlim,ksec,dist,knmay,klen: real
  kx,ky,top,tcl: array(nodos) of real
  sol: dynamic array(ptos,ptos) of real
  kmaxclust = 3
  cdist: array(ptos,ptos) of real
  zx,zy: array(ptos) of real
  ztop,ztcl: array(ptos) of integer
  zclus: array(ptos) of integer
  zgroup: array(ptos) of integer
  w: dynamic array(ptos,ptos) of mpvar
  trinp,trout,trsec,trot,trct:array(nodos) of integer
  rut3: dynamic array(ptos) of integer
  arco: dynamic array(ptos,ptos) of integer
end-declarations

! VERSION ORIGINAL PARA EL MANEJO DEL TW SENCILLO
! INCLUYE 4 FASES DE DEPURACION
! XPRS_BAR, XPRS_PRI, XPRS_DUAL

starttime := gettime

fopen("TSPTW100.TXT",F_INPUT)
kn:= 1
repeat
  read(ki,zx(kn),zy(kn),ztop(kn),ztcl(kn))
  kn := kn + 1
until (getparam('nbread') < 5)
fclose(F_INPUT)

forall(i in ptos, k in ptos) do
  if i <> k then
    cdist(i,k) := round(((zx(i) - zx(k))^2 +
      (zy(i) - zy(k))^2)^0.5)
  else
    cdist(i,k) := 1e6
  end-if
end-do
```

```

fopen("CLUSTER.TXT",F_INPUT)
repeat
  read(kn,zclus(kn))
  if getparam('nbread') >= 2 then
    zgroup(zclus(kn)) := zgroup(zclus(kn)) + 1
  end-if
until (getparam('nbread') < 2)
fclose(F_INPUT)

fopen("TOUR.TXT",F_INPUT)
kn := 1
repeat
  read(trinp(kn),trout(kn),trsec(kn),trot(kn),trct(kn))
  kn := kn + 1
until (getparam('nbread') < 5)
fclose(F_INPUT)

kn := 0
forall(j in ptos) do
  if zgroup(j) >= 1 then
    kn:= kn + 1
    ztrad(kn) := j
  end-if
end-do

forall(j in nodos) do
  ar1 := ztrad(trinp(j))
  ar2 := ztrad(trout(j))
  forall(i in ptos, k in ptos) do
    if (zclus(i) = ar1 and zclus(k) = ar1) or
      (zclus(i) = ar1 and zclus(k) = ar2) then
      create(w(i,k))
      w(i,k) is_binary
    end-if
  end-do
end-do

forall(i in ptos) sum(j in ptos) w(i,j) = 1
forall(j in ptos) sum(i in ptos) w(i,j) = 1
distancia:= sum(i in ptos, j in ptos) cdist(i,j)*w(i,j)

! INICIA PROCESO PARA ELIMINACION DE SUB-TOURS
kc :=1
repeat
  minimize(kmotor,distancia)
  ksol := getobjval
  savebasis(1)

  forall(i in ptos, j in ptos) sol(i,j):= getsol(w(i,j))

  kmay := 0
  kmen := kred
  kr := 0
  repeat
    forall(i in ptos) rut3(i) := 0

```



```

ki:= 1
kp1:=0
repeat
    kj:= 1
    repeat
        if sol(ki,kj) = 1 then
            kp1:=ki
            break 2
        end-if
        kj := kj + 1
    until kj > kred
    ki := ki + 1
until ki > kred

if kp1 = 0 then
    break      ! se acabaron los sub-tours
end-if

kp1 := 1
rut3(ki) := 1
kj := 1
kr := kr + 1
repeat
    if sol(ki,kj) = 1 then
        kp1 := kp1 + 1
        rut3(kj):= 1
        sol(ki,kj) := 0
        ki := kj
        kj := 0
    end-if
    kj := kj + 1
until kj > kred
kp1 := kp1 - 1

if kp1 <= kmen and kp1 >= 2 then
    kmen := kp1
end-if
if kp1 >= kmay then
    kmay:= kp1
end-if

if kp1 < kred then
    ! SE AGREGA EL SUB-TOUR ELIMINATION CONSTR
    sum(k1 in ptos, k2 in ptos |
        rut3(k1) = 1 and rut3(k2) = 1 )
        w(k1,k2) <= kp1-1
    end-if
until kp1 <= 0 or kp1 >= kred
if kp1 = kred then
    break
end-if
loadprob(distancia)
loadbasis(1)

kc := kc + 1

```

```

        writeln (ksol, " = ", kmen, ",", kmay, ",", kr, " Ti= ", gettime-
starttime)
    until false

    minimize(kmotor,distancia)
    ksol := getobjval
    writeln ("*** Termina:", ksol)

    forall(i in ptos, j in ptos) sol(i,j) := getsol(w(i,j))
    ki := ztrad(trinp(1))
    kj := 1
    ktw := 0
    repeat
        if sol(ki,kj) <> 0 then
            ktw := ktw + 1
            writeln(ki,",",kj,",",ktw,",",ztop(kj),",",ztc1(kj))
            sol(ki,kj) := 0
            ki := kj
            kj := 0
        end-if
        kj := kj + 1
    until kj > kred
    writeln("Tiempo: ",gettime-startime)

    fopen("RUTA.TXT",F_OUTPUT)
    forall(i in ptos, j in ptos) sol(i,j) := getsol(w(i,j))
    ki := ztrad(trinp(1))
    kj := 1
    ktw := 0
    repeat
        if sol(ki,kj) <> 0 then
            ktw := ktw + 1
            writeln(ki,",",kj,",",ktw,",",ztop(kj),",",ztc1(kj))
            sol(ki,kj) := 0
            ki := kj
            kj := 0
        end-if
        kj := kj + 1
    until kj > kred
    fclose(F_OUTPUT)

end-model

```

12. Referencias Bibliográficas.

- Aguerrondo, Inés. (1998), *La Educación del tercer milenio*, “Revista Academia”, Argentina.
- Alander, Jarmo. (2003), *An Indexed Bibliography of Genetic Algorithms in Logistics*, Department of Engineering, University of Vaasa, Finland.
- Applegate, D; Bixby, R; Chvátal, V; (1998), *On the solution of traveling salesman problems*. “Documenta Mathematica Extra Volume ICM III”, EUA.
- Applegate, David. (1995), *Finding Cuts in the TSP*, Princeton University, EUA.
- Arsham, Hossein. (1996), *Applied Management Science*, Cambridge University Press, EUA.
- Ascheuer, N; Fischetti, M; Grotschel, M. (2001), *Solving ATSP with time windows by branch-and-cut*, Springer-Verlag, EUA.
- Ascheuer, N; Jünger, M; Reinelt, G. (2000), *A branch & cut algorithm for the asymmetric Traveling Salesman Problem with precedence constraints*, “Computational Optimization and Applications 17(1)”, EUA.
- Baker, Kenneth. (1999), *Gaining Insight in Linear Programming*, “Informs”, Vol.1, EUA.
- Balas, E; Simonetti, N. (1996), *Linear time dynamic programming algorithms for some new classes of restricted TSP's*, “Management Science Research Report 617”, Graduate School of Industrial Administration, Carnegie Mellon University, EUA.
- Bednarek, A. (1984), *Analogies between Analogies: The mathematical reports of S.M. Ulam and his Los Alamos collaborators*, University of California Press, EUA.
- Bellman, Richard. (1957), *Dynamic Programming*, Princeton University Press, Princeton Nueva Jersey.
- Berry, William. (1990), *Management Decision Science*, 1thEd, Irwin, EUA.
- Bixby, Robert. (1999), *MIP: theory and practice closing the gap*, Ilog Cplex Division, Department of Computational and Applied Mathematics Rice University, Houston EUA.
- Bremermann, H. (1962), *Optimization through Evolution and Recombination*, Spartan Books, Alemania.
- Bryne, John. (1997), *Los Nuevos gurúes de la Administración*, “Revista Gestión y Estrategia”, No.11-12, México.
- Buckles, Bill. (1992), *Genetic Algorithms*, IEEE Computer Society Press, EUA.
- Cheng, R. (1997), *Genetic Algorithms and Engineering Design*, John Wiley Interscience, Boston Ma, EUA.

- Chowdury, M. (1997), *Messy Genetic Algorithms based new learning method for fussy controllers*, “Proceedings of the IEEE International Conference on Industrial Technology”, The MIT Press, Shanghai China.
- Christofides, N. (1976), *Worst-case analysis of a new heuristic for the traveling salesman problem*, “Report 388 Graduate School of Industrial Administration”, Carnegie-Mellon University, Pittsburgh EUA.
- Cochran, W. (1977), *Experimental Design*, John Wiley & Sons, New York.
- Coello, Carlos. (1995), *Introducción a los Algoritmos Genéticos*, “Tecnologías de Información y Estrategias de Negocios, No. 17”, México.
- Cook, W; Rich, Jennifer. (1999), *A parallel cutting-plane algorithm for the vehicle routing problem with time windows*, Computational and Applied Mathematics Rice University, Houston EUA.
- Dale, Mann. (1999), *Achievement gains from a Statewide Comprehensive Instructional Program*, Milken Family Foundation, EUA.
- Dávila, S. (1999), *El papel del docente en la calidad educativa*, México, en <http://www.nalejandria.com/akademeia/sdavila/>
- Deb, K. (1995), *Simulated Binary crossover in evolutionary optimization*, “Complex Systems No. 9”, EUA.
- Delgado, Mónica. (2001), *SEP difunde resultados de la evaluación Internacional: Entrevista al Secretario de Educación Reyes Taméz*, “Periódico el Norte”, México.
- Dell 'Amico, M; Martello, S. (1997), *Linear assignment: Annotated Bibliographies in Combinatorial Optimization*, Wiley & Sons editors, EUA.
- Desrosiers, J; Dumas, Y; Solomon, M; Soumis, F. (1995), *Time constrained routing and scheduling*, G.L. Nemhauser editor, Handbooks in Operations Research and Management Science, Holanda.
- Drezner, Zvi. (1996), *Facility Location: A Survey of Applications and Methods*, 1st Ed, Springer Verlag, EUA.
- Dumas, Y; Desrosiers, J; Gelinas, E; Solomon, M. (1995), *An algorithm for the traveling salesman problem with time windows*, “Operations Research 43(2)”, EUA.
- Eijl Van, C. (1995), *A polyhedral approach to the delivery man problem*, “Technical Report 95–19”, Department of Mathematics and Computer Science, Eindhoven University of Technology, The Neatherlands.
- Encuesta Nacional de Empleo. (2001), *Base de Datos y Tabulados de la Muestra Censal*, “XII Censo General de Población y Vivienda”, México.

- Esbensen, H. (1995), *Computing near-optimal solutions to the Steiner problem using genetics algorithms*, Networks Interscience, Vol 26, No.4, EUA.
- Fisher, Marshall. (1995), *Overview over optimization models in transportation*, Handbooks in Operations Research and Management Science, North Holland, EUA.
- Fisher, R. (1971), *The Design of Experiments*, Hafner Press & Macmillan Publishers, London.
- Focacci, F; Milano, M. (1999), *Solving Tsp with time windows with constraints*, "INFORMS Journal of Computing, ICLP 99", International Conference on Logic Programming, EUA.
- Focacci, Filippo; Milano, Michela. (2000), *Solving TSP with Time Windows with Constraints*, Dip. Ingegneria, University of Ferrara, Italia.
- Fuente Guía Roji. (2000), *Censo de empresas del área metropolitana de la ciudad de Monterrey*, México.
- Garey, D. (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, New York.
- Garey, M; Johnson, D. (1989), *Computers and Intractability: a Guide to the Theory of NP-Completeness*, Freeman Ed, San Francisco EUA.
- Geoffrion, Arthur. (1999), *The decadence of operations research*, ORMS Today, EUA.
- Geoffrion, Arthur. (2000), *The value of operations research*, "ORMS Today", EUA.
- Glover, Fred. (1986), *Future paths for integer programming and links to artificial intelligence*, "Computers and Operations Research", EUA.
- Glover, Fred. (1990), *Tabu Search: A Tutorial*, "Interfaces, Vol 20, No. 4", EUA.
- Glover, Fred. (1993), *A user's guide to Tabu Search*, "Annals of Operations Research, Vol. 41", EUA.
- Goldberg, D. (1995), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley Pub Co, University of Massachusetts, EUA.
- Goldberg, D. (1996), *Don't worry, be messy*, "Proceedings of the 9th International Conference on Genetic Algorithms", The MIT Press, Cambridge Mass, EUA.
- Goldberg, David. (1989), *Genetic Algorithms in Optimization and Machine Learning*, Addison-Wesley Publishing Company, EUA.
- Gomez, M. (1998), *Inserción ocupacional de los egresados universitarios*, "IV Congreso Nacional de Estudios del Trabajo", Argentina.
- Gomory, R. (1963), *An algorithm for integer solutions to linear programs*, Recent Advances in

- Mathematical Programming (R. L. Graves and P. Wolfe, eds.), McGraw-Hill, New York.
- Greenberg, H. (1996), *Mathematical Programming Glossary*, disponible en <http://www.cudenver.edu/~hgreenbe/glossary>.
- Hartmann, Sonke. (2000), *Project Scheduling Under Limited Resources: Models, Methods, and Applications*, Springer Verlag, EUA.
- Hicks, C. (1994), *Fundamental Concepts in the Design of Experiments*, Holt and Winston Ed, New York.
- Hinkelmann, K. (1994), *Design and Analysis of Experiments*, McMillan, EUA.
- Hoffmeister, F. (1992), *Genetic Algorithms and Evolution Strategies: Similarities and Differences*, "Technical Report No.1", University of Dortmund, Alemania.
- Holland, John. (1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press, EUA.
- Hollstein, R. (1991), *Artificial Genetic adaptation in computer control systems*, Phd Thesis, University of Michigan, EUA.
- Hsiang, Thomas. (2001), *The Illusion of Power*, "ORMS Today", EUA.
- Hunter, W. (1986), *Statistics for Experimenters*, John Willey & Sons, EUA.
- Jonker, R; Volgenant, T. (1982), *A branch and bound algorithm for the symmetric traveling salesman problem*, "European Journal of Operational Research, No. 2", Dinamarca.
- Jünger, M; Rinaldi, G; Thienel, S. (2000), *Practical performance of efficient minimum cut algorithms*, "Algorithmica No. 26", Dinamarca.
- Karger, D. (1993), *Global min-cuts in RNC and other ramifications of a simple min-cut algorithm*, "Proceedings of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms", EUA.
- Klein, Robert. (1999), *Scheduling of Resource-Constrained Projects*, Kluwer Academic, EUA.
- Koza, John. (1992), *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, The MIT Press, EUA.
- Laguna, Manuel. (1994), *A guide to implementing Tabu Search*, "Technical Report, Graduate School of Business, University of Colorado", Boulder Colorado.
- Land, A; Doig, H. (1960), *An Automatic Method for Solving Discrete Programming Problems*, Econometrica, EUA.
- Lee, R. (1976), *Worst-case analysis of a new heuristic for the travelling salesman problem*, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh PA. Disponible en <http://www.ms.uky.edu/~jlee/jlsup/jlsup.html>.
- Lenstra, J. (1985), *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, Ediitorial Wiley, Chichester England.

- Lenstra, K. (1990), *A Variable Depth Approach for the Single-Vehicle Pickup and Delivery Problem with Time Windows*, "COSOR No. 90-48", Eindhoven University of Technology, Alemania.
- Levine, G. (1997), *Developing fitter genetic algorithms*, INFORMS Journal of Computing, Vol. 9, No. 5, EUA.
- Lorenzen, T. (1993), *Design of Experiments*, Marcel Dekker Inc, EUA.
- Marchand, Horacio. (2001), *¿Una nación en Riesgo?*, EGADE ITESM, México.
- Mathias, K. (1994), *Transforming the search space with gray coding*, Proceedings of the IEEE International Conference on Evolution Computation, Piscataway New Jersey, EUA.
- Mingozzi, A; Bianco, L; Ricciardelli, S. (1997), *Dynamic programming strategies for the travelling salesman problem with time windows and precedence constraints*, "Operations Research No. 45", EUA.
- Mitrovic, Snezana. (1998), *Pickup and Delivery Problem with Time Windows*, "Technical Report SFU CMPT TR 1998-12", Canada.
- Mittelmann, H. (2002), *Decision Tree for Optimization Software*, disponible en <http://plato.la.asu.edu/guide.html>.
- Montgomery, D. (1991), *Diseño y análisis de experimentos*, Grupo Editorial Ibero-Américo, México.
- Moore, Lawrence. (1993), *Management Science*, 4thEd, Allyn and Bacon, EUA.
- Mota, Flavio. (1998), *El deber ser y el querer ser en la Educación*, "Revista Academia", México, disponible en <http://kepler.uag.mx/temasedu/deberser.htm>.
- Optima. (1998), *Mathematical Programming Society Newsletter*, EUA.
- Padberg, M; Rinaldi, G. (1991), *A Branch and cut algorithm for the resolution of large-scale symmetric traveling salesman problem*, "SIAM Review No. 33", EUA.
- Palmgren, Myrna. (2001), *A Column Generation Algorithm for the Log Truck Scheduling Problem*, Department of Science and Technology (ITN), Linköping University, Norrköping Sweden.
- Parker, R; Rardin, R. (1998), *Discrete Optimization*, Academic Press, New York EUA.
- Pawda, Juan. (2000), *Modelos de Investigación de Operaciones*, 1era.Ed, Limusa, México.
- Premkumar, G. (1999), *Telecommunications Network design: a genetic algorithm approach*, School of Information Sciences and Technology & Pennsylvania State College of Business Administration, EUA.
- Quintana, Enrique. (2001), *Matemáticas y Economía*, "Editorial el Norte", México.
- Render, Barry. (2000), *Quantitative Analysis for Management*, 7thEd, Prentice Hall, EUA.

- Reeves, C. (1995), *Genetics algorithms flow shop sequencing*, Computers & Operations Research, Vol 22, EUA.
- Ríos, Roger. (1999), *Aplicaciones del TSP*, “Ingenierías 2(4)”, México.
- Rodríguez, Joaquin. (1998), *Introd. a la Administración con Enfoque de Sistemas*, 3era Ed, ECAFSA, México.
- Savelsberg, M. (1995), *Local search in Routing Problem with Time Windows*, Annals of Operations Research, Rotherdam Holanda.
- Savelsbergh, M. (1998), *Local Search in Physical Distribution Management*, Eindhoven University of Technology, Rotterdam Holanda.
- Skiena, S. (1990), *Implementing Discrete Mathematics: Combinatorics and Graph Theory in Mathematica*, Addison-Wesley, Redwood CA. Disponible en <ftp://ftp.cs.sunysb.edu/pub/Combinatorica/>
- Sodhi, ManMohan. (2001), *A Match Made in Heaven*, “ORMS Today”, EUA.
- Solomon, M. (1984), *On the worst-case performance of some heuristics for the vehicle routing and scheduling problem with time window constraints*, “Report 83-05-03”, The Wharton School, University of Pennsylvania EUA.
- Srinivas, M. (1999), *Genetic Algorithms: A Survey*, IEEE Computer, EUA.
- Stewardson, D. (2002), *Overcoming Complexity: optimizing genetic algorithms for use in complex scheduling problems via designed experiments*, University of Newcastle, United Kingdom.
- Streifel, R. (1999), *Dynamic Fussy control of genetic algorithm parameters*, IEEE Trans. Syst., EUA.
- Tenti, Emilio. (2000), *Educación Participativa*, Argentina, disponible en <http://www.utdt.edu/eduforum/>.
- Toutenggburg, H. (1995), *Experimental Design and Model Choice*, Physica-Verlag, EUA.
- Tsitsiklis, J. (1992), *Special cases of traveling salesman and repairman problems with time windows*, “Networks No. 22”, EUA.
- Turing, A. (1950), *Computer Machinery and Intelligence*, MIND, EUA.
- Weber, D. (2000), *A First Course in the Design of Experiments*, CRC Press, EUA.
- Whitley, D. (1998), *Representation issues in Neighborhood Search and Evolutionary Algorithms*, John Wiley and Sons, University of Sussex, England, Capítulo 3.