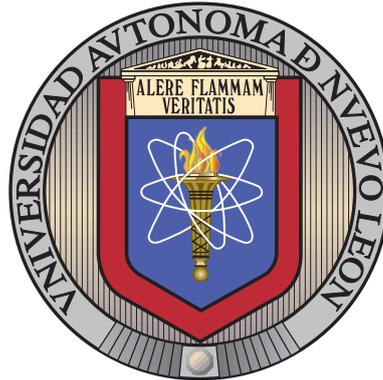


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO



PROBLEMA DEL AGENTE VIAJERO SELECTIVO  
CON RESTRICCIONES ADICIONALES

POR

PAMELA JOCELYN PALOMO MARTÍNEZ

COMO REQUISITO PARA OBTENER EL GRADO DE

MAESTRÍA EN CIENCIAS

EN INGENIERÍA DE SISTEMAS

JUNIO, 2015

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO



PROBLEMA DEL AGENTE VIAJERO SELECTIVO  
CON RESTRICCIONES ADICIONALES

POR

PAMELA JOCELYN PALOMO MARTÍNEZ

COMO REQUISITO PARA OBTENER EL GRADO DE

MAESTRÍA EN CIENCIAS

EN INGENIERÍA DE SISTEMAS

JUNIO, 2015

**Universidad Autónoma de Nuevo León**  
**Facultad de Ingeniería Mecánica y Eléctrica**  
**Subdirección de Estudios de Posgrado**

Los miembros del Comité de Tesis recomendamos que la Tesis «Problema del Agente Viajero Selectivo con Restricciones Adicionales», realizada por la alumna Pamela Jocelyn Palomo Martínez, con número de matrícula 1502428, sea aceptada para su defensa como opción al grado de Maestría en Ciencias en Ingeniería de Sistemas.

El Comité de Tesis



---

Dra. María Angélica Salazar Aguilar

Asesor



---

Dr. Vincent André Lionel Boyer

Revisor

---

Dr. José Luis González Velarde

Revisor

Vo. Bo.

---

Dr. Simón Martínez Martínez

Subdirector de Estudios de Posgrado

*A Oscar.*

# ÍNDICE GENERAL

---

<b>Agradecimientos</b>	<b>xv</b>
<b>Resumen</b>	<b>xvii</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Problemas de ruteo de vehículos con beneficios . . . . .	2
1.2. Motivación de la tesis . . . . .	4
1.3. Metodología de investigación . . . . .	5
1.4. Estructura de la tesis . . . . .	6
<b>2. Planteamiento del problema</b>	<b>8</b>
2.1. Estado del arte del Problema del Agente Viajero Selectivo . . . . .	8
2.2. Estado del arte del Problema del Agente Viajero Selectivo con uno o múltiples agentes y restricciones de visitas obligatorias y conflictos entre nodos . . . . .	11
2.3. Descripción formal del problema . . . . .	13
2.4. Formulación matemática del Problema del Agente Viajero Selectivo con restricciones adicionales . . . . .	14

---

2.4.1. Modelo matemático . . . . .	14
2.4.2. Restricciones de eliminación de ciclos para el OP . . . . .	18
<b>3. Comparaciones entre los modelos propuestos</b>	<b>22</b>
3.1. Dimensión de los modelos propuestos . . . . .	22
3.2. Entorno experimental . . . . .	24
3.3. Descripción de las instancias de prueba . . . . .	25
3.4. Soluciones obtenidas por los modelos propuestos . . . . .	26
3.5. Relajación lineal de los modelos propuestos . . . . .	27
<b>4. Algoritmo heurístico propuesto</b>	<b>30</b>
4.1. Greedy Randomized Adaptive Search Procedure . . . . .	31
4.1.1. Fase de construcción . . . . .	31
4.1.2. Fase de postprocesamiento . . . . .	32
4.1.3. GRASP reactivo . . . . .	33
4.2. Variable Neighborhood Search . . . . .	34
4.2.1. Variable Neighborhood Descent . . . . .	34
4.2.2. General Variable Neighborhood Search . . . . .	35
4.3. Algoritmo híbrido tipo GRASP con VNS para el OP con restricciones adicionales . . . . .	37
4.3.1. Construcción de soluciones . . . . .	38
4.3.2. Filtrado de soluciones . . . . .	42
4.3.3. Vecindarios utilizados en el VNS . . . . .	43

---

4.3.4.	Recuperación de factibilidad . . . . .	45
4.3.5.	Búsqueda local por VND . . . . .	46
<b>5.</b>	<b>Resultados computacionales del algoritmo heurístico</b>	<b>52</b>
5.1.	Orden de la ejecución de los algoritmos de búsqueda local en el VND	53
5.1.1.	Tamaño del vecindario N1 . . . . .	53
5.1.2.	Tamaño del vecindario N2 . . . . .	54
5.1.3.	Orden de los algoritmos de búsqueda local . . . . .	54
5.2.	Orden de la exploración de los vecindarios de agitación en el VNS . .	55
5.3.	Parámetros utilizados en el GRASP . . . . .	59
5.4.	Resultados computacionales . . . . .	60
<b>6.</b>	<b>Procedimiento de generación de columnas propuesto</b>	<b>79</b>
6.1.	Aspectos teóricos del método de generación de columnas . . . . .	80
6.2.	Uso de métodos exactos en problemas relacionados . . . . .	82
6.3.	Formulación matemática del OP con restricciones adicionales como un problema de gran escala . . . . .	85
6.3.1.	Problema maestro . . . . .	85
6.3.2.	Subproblema . . . . .	86
6.4.	Algoritmo de generación de columnas para el OP con restriccciones adicionales . . . . .	89
6.4.1.	Inicialización . . . . .	89
6.4.2.	Solución del problema maestro . . . . .	89

---

6.4.3. Solución del subproblema . . . . .	89
6.4.4. Cálculo de cotas . . . . .	91
6.5. Resultados computacionales . . . . .	92
6.5.1. Comparación de las cotas obtenidas contra el valor óptimo . . . . .	95
<b>7. Conclusiones y trabajo futuro</b>	<b>97</b>
7.1. Conclusiones . . . . .	97
7.2. Trabajo futuro . . . . .	98
<b>A. Tablas de resultados</b>	<b>100</b>
A.1. Tablas referentes a pruebas entre los modelos . . . . .	100
A.2. Tablas referentes a los resultados del híbrido GRASP-VNS . . . . .	120
A.3. Tablas referentes a los resultados del método de generación de columnas	130
<b>B. Generación de instancias</b>	<b>136</b>
B.1. Generación de conflictos en nivel alto . . . . .	136
B.2. Generación de conflictos en nivel bajo . . . . .	138
B.3. Generación de nodos obligatorios . . . . .	138
B.4. Generación de la máxima duración de la ruta . . . . .	140

# ÍNDICE DE FIGURAS

---

3.1. Gráfica de la diferencia entre el número de restricciones de los modelos	23
3.2. Gráfica de la diferencia entre el número de variables de los modelos	24
3.3. Comportamiento de las relajaciones lineales de los modelos propuestos	29
4.1. Vecindarios utilizados en el VNS	45
4.2. Recuperación de factibilidad debido a la violación de la duración máxima	46
4.3. Recuperación de factibilidad debido a existencia de conflictos	46
5.1. Intervalos de confianza para la media del gap de cada versión del GRASP-VNS con una confiabilidad del 95 %	58
5.2. Intervalos de confianza para la media del mejor gap en cada clase con una confiabilidad del 95 %	62
5.3. Intervalos de confianza para la media del peor gap en cada clase con una confiabilidad del 95 %	62
5.4. Intervalos de confianza para la media del tiempo de ejecución en cada clase con una confiabilidad del 95 %	64
6.1. Intervalos de confianza para el gap entre cotas en cada clase con una confiabilidad del 95 %	95

# ÍNDICE DE TABLAS

---

1.1. Problemas de ruteo de vehículos con beneficios . . . . .	4
2.1. Estado del arte del OP y el TOP con visitas obligatorias y conflictos .	13
3.1. Comparación entre las dimensiones de los modelos . . . . .	23
3.2. Descripción de las características de los conjuntos de instancias de prueba . . . . .	25
3.3. Comparaciones entre las instancias resueltas por las formulaciones propuestas . . . . .	26
3.4. Comparaciones entre las instancias resueltas por las relajaciones lineales de las formulaciones propuestas . . . . .	27
5.1. Versiones del algoritmo GRASP-VNS . . . . .	55
5.2. Prueba de Kruskal Wallis: Gap de la función objetivo contra versión GRASP-VNS . . . . .	57
5.3. Prueba de Kruskal Wallis: Gap del tiempo contra versión GRASP-VNS	57
5.4. Intervalos de confianza para la media del gap de cada versión del GRASP-VNS con una confiabilidad del 95 % . . . . .	58
5.5. Comparación de la calidad de las soluciones obtenidas por el algoritmo según distintos criterios de paro . . . . .	60

---

5.6. Resultados del algoritmo híbrido GRASP-VNS . . . . .	61
5.7. Prueba de Kruskal Wallis: Tiempo de ejecución contra clase de instancias . . . . .	63
5.8. Intervalos de confianza para la media del gap de cada versión del GRASP-VNS con una confiabilidad del 95 % . . . . .	64
5.9. Comparaciones entre los resultados obtenidos mediante la heurística GRASP-VNS y las mejores soluciones conocidas de los modelos M1 y M2 en la Clase 1 . . . . .	66
5.10. Comparaciones entre los resultados obtenidos mediante la heurística GRASP-VNS y las mejores soluciones conocidas de los modelos M1 y M2 en la Clase 2 . . . . .	69
5.11. Comparaciones entre los resultados obtenidos mediante la heurística GRASP-VNS y las mejores soluciones conocidas de los modelos M1 y M2 en la Clase 3 . . . . .	71
5.12. Comparaciones entre los resultados obtenidos mediante la heurística GRASP-VNS y las mejores soluciones conocidas de los modelos M1 y M2 en la Clase 4 . . . . .	73
5.13. Comparaciones entre los resultados obtenidos mediante la heurística GRASP-VNS y las mejores soluciones conocidas de los modelos M1 y M2 en la Clase 5 . . . . .	75
5.14. Comparaciones entre los resultados obtenidos mediante la heurística GRASP-VNS y las mejores soluciones conocidas de los modelos M1 y M2 en la Clase 6 . . . . .	77
6.1. Métodos exactos utilizados en problemas afines . . . . .	84
6.2. Frecuencia y fechas de uso de métodos exactos para problemas afines	85

---

6.3. Resultados obtenidos por el método de generación de columnas . . . .	93
6.4. Prueba de Kruskal Wallis: Gap entre cotas contra clase de instancias	94
6.5. Intervalos de confianza para la media del gap entre cotas en cada clase con una confiabilidad del 95 % . . . . .	94
6.6. Gap de las cotas con respecto al óptimo . . . . .	96
A.1. Resultados obtenidos por Cplex 12.6 haciendo uso de los modelos propuestos en las instancias de la clase 1 . . . . .	102
A.2. Resultados obtenidos por Cplex 12.6 haciendo uso de los modelos propuestos en las instancias de la clase 2 . . . . .	104
A.3. Resultados obtenidos por Cplex 12.6 haciendo uso de los modelos propuestos en las instancias de la clase 3 . . . . .	105
A.4. Resultados obtenidos por Cplex 12.6 haciendo uso de los modelos propuestos en las instancias de la clase 4 . . . . .	107
A.5. Resultados obtenidos por Cplex 12.6 haciendo uso de los modelos propuestos en las instancias de la clase 5 . . . . .	108
A.6. Resultados obtenidos por Cplex 12.6 haciendo uso de los modelos propuestos en las instancias de la clase 6 . . . . .	110
A.7. Resultados obtenidos por Cplex 12.6 haciendo uso de la relajación lineal de los modelos propuestos en las instancias de la clase 1 . . . .	112
A.8. Resultados obtenidos por Cplex 12.6 haciendo uso de la relajación lineal de los modelos propuestos en las instancias de la clase 2 . . . .	113
A.9. Resultados obtenidos por Cplex 12.6 haciendo uso de la relajación lineal de los modelos propuestos en las instancias de la clase 3 . . . .	114

A.10.Resultados obtenidos por Cplex 12.6 haciendo uso de la relajación lineal de los modelos propuestos en las instancias de la clase 4 . . . . .	116
A.11.Resultados obtenidos por Cplex 12.6 haciendo uso de la relajación lineal de los modelos propuestos en las instancias de la clase 5 . . . . .	118
A.12.Resultados obtenidos por Cplex 12.6 haciendo uso de la relajación lineal de los modelos propuestos en las instancias de la clase 6 . . . . .	119
A.13.Resultados obtenidos el híbrido GRASP-VNS en las instancias de la clase 1 . . . . .	121
A.14.Resultados obtenidos el híbrido GRASP-VNS en las instancias de la clase 2 . . . . .	123
A.15.Resultados obtenidos el híbrido GRASP-VNS en las instancias de la clase 3 . . . . .	124
A.16.Resultados obtenidos el híbrido GRASP-VNS en las instancias de la clase 4 . . . . .	126
A.17.Resultados obtenidos el híbrido GRASP-VNS en las instancias de la clase 5 . . . . .	128
A.18.Resultados obtenidos el híbrido GRASP-VNS en las instancias de la clase 6 . . . . .	129
A.19.Resultados obtenidos por el método de generación de columnas en las instancias de la clase 1 . . . . .	131
A.20.Resultados obtenidos por el método de generación de columnas en las instancias de la clase 2 . . . . .	131
A.21.Resultados obtenidos por el método de generación de columnas en las instancias de la clase 3 . . . . .	132

---

A.22.Resultados obtenidos por el método de generación de columnas en las instancias de la clase 4 . . . . .	133
A.23.Resultados obtenidos por el método de generación de columnas en las instancias de la clase 5 . . . . .	134
A.24.Resultados obtenidos por el método de generación de columnas en las instancias de la clase 6 . . . . .	135

# AGRADECIMIENTOS

---

Esta tesis es el fruto de dos años de trabajo, durante los cuales tuve la bendición de contar con el apoyo de personas e instituciones que merecen mi sincero agradecimiento por ello.

Agradezco a la Universidad Autónoma de Nuevo León, a la Facultad de Ingeniería Mecánica y Eléctrica y al Programa de Posgrado en Ingeniería de Sistemas por haberme acogido como alumna de maestría; asimismo, reconozco al Consejo Nacional de Ciencia y Tecnología por beneficiarme con una beca que me permitió realizar mis estudios de posgrado y con una beca para realizar una estancia de investigación en el extranjero.

Quiero reconocer a mi directora de tesis, la Dra. Angélica Salazar, porque antes de ser mi asesora, fue una excelente profesora y, como directora de tesis, sus comentarios y consejos fueron de gran ayuda para mejorar la calidad de mi trabajo. Le doy las gracias por confiar en mí como estudiante, por motivarme siempre a enfrentar retos cada vez mayores y por haber hecho las veces de profesora, asesora y, algunas veces, hasta psicóloga.

Doy gracias a los revisores de esta tesis, el Dr. Vincent Boyer y el Dr. José Luis González Velarde, por haber dedicado parte de su tiempo a la lectura de este trabajo y por sus acertados comentarios.

Agradezco a la Universidad Técnica Federico Santa María y, en particular, al Dr. Victor Albornoz, por haberme recibido para hacer una estancia de investigación, experiencia que influyó en gran medida en el resultado final de la tesis y de la cual

conservo gratos recuerdos.

Quiero agradecer a mis compañeros de generación por todas las veces que recibí ayuda de parte de ellos, por todas las experiencias que compartimos y por la amistad que forjamos.

Doy gracias a Oscar Perez por escuchar mis presentaciones antes de cada congreso o seminario, por prestarme atención cada vez que me emociono por algo nuevo que aprendí (aunque a veces no sepa de qué estoy hablando exactamente), por esperarme en cada viaje, por resolver mis problemas técnicos y, sobre todo, por amarme. Por todo, gracias.

Agradezco a mi madre y a mis hermanos por creer en mí siempre y por todo el apoyo que me han brindado desde que salí de casa para estudiar la licenciatura hace algunos años atrás.

# RESUMEN

---

Pamela Jocelyn Palomo Martínez.

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

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

Título del estudio:

## PROBLEMA DEL AGENTE VIAJERO SELECTIVO CON RESTRICCIONES ADICIONALES

Número de páginas: 152.

**OBJETIVOS Y MÉTODO DE ESTUDIO:** En esta tesis se introduce una variante del Problema del Agente Viajero Selectivo, también conocido en la literatura como *Orienteering Problem* (OP). En el OP se tiene un conjunto de clientes potenciales, a cada uno de los cuales se le asocia una puntuación o beneficio que recibe el agente al visitarlo, el objetivo es el de diseñar una ruta que comience y termine en el depósito y que maximice el puntaje colectado, tomando en cuenta que existe un límite máximo en la duración de la ruta. En este trabajo se consideran restricciones de conflictos entre clientes, es decir, si dos de ellos tienen conflicto, no pueden ser incluidos ambos en la ruta; por otra parte, existe un subconjunto de clientes que deben ser visitados de manera obligatoria.

Se proponen dos modelos matemáticos del problema, cuya diferencia principal es la manera en que aborda la eliminación de ciclos. El primer modelo usa restricciones de tipo secuencial inspiradas en las propuestas por Miller *et al.* (1960) y el segundo utiliza restricciones basadas en flujo de múltiples productos y se basan en las restricciones propuestas por Wong (1980) y Claus (1984).

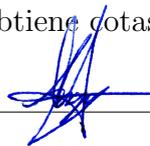
Asimismo, se proponen dos algoritmos para la solución del problema planteado, el primero es de tipo heurístico y está basado en un esquema GRASP (Greedy Randomized Adaptive Search Procedure) reactivo, cuya fase de mejora es un método tipo VNS (Variable Neighborhood Search) general, el segundo es una estrategia de descomposición basada en generación de columnas.

El desempeño de los algoritmos propuestos es evaluado a través de experimentos computacionales sobre un gran conjunto de instancias y los resultados obtenidos son comparados contra las soluciones óptimas obtenidas al resolver los modelos matemáticos haciendo uso del solver Cplex 12.6.

CONTRIBUCIONES Y CONCLUSIONES: Esta tesis introduce una variante del *Orienteering Problem* que no había sido estudiada en la literatura. Como resultado de este trabajo se presentan dos formulaciones matemáticas para el problema y dos algoritmos para la solución del mismo.

El algoritmo heurístico propuesto obtiene soluciones cercanas al óptimo en la mayoría de las instancias para las cuales este es conocido y el método de generación de columnas propuesto obtiene cotas inferiores de alta calidad.

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



Dra. María Angélica Salazar Aguilar

## CAPÍTULO 1

# INTRODUCCIÓN

---

En esta tesis se presenta una variante del Problema del Agente Viajero Selectivo, también conocido en la literatura como *Orienteering Problem* (OP), en el cual se tiene un conjunto de clientes, cada uno con una puntuación no negativa asociada; el objetivo es maximizar la puntuación total colectada al visitar un subconjunto selecto de clientes, tomando en cuenta que la duración de la ruta no debe exceder un límite máximo. En este trabajo se consideran, además, restricciones de visitas obligatorias a clientes y de conflictos entre visitas, es decir, una ruta no puede visitar a dos o más clientes con conflictos entre ellos.

Se han considerado estas restricciones adicionales, debido a que contribuyen a una representación matemática más precisa de algunas problemáticas reales, como lo son el diseño de rutas para empresas dedicadas a la recolección de residuos peligrosos, en el cual se debe considerar que existen residuos que no pueden ser transportados en la misma unidad debido a que pueden presentarse reacciones peligrosas entre ellos; y la planificación de actividades de mantenimiento vial, en la que existen actividades que deben realizarse de manera obligatoria para asegurar el correcto flujo vehicular y existen otras que, de realizarse en el mismo día, provocarían la interrupción del tráfico (Salazar-Aguilar *et al.*, 2014).

Se proponen dos algoritmos para resolver el problema planteado, el primero está basado en un esquema GRASP (Greedy Randomized Adaptive Search Procedure) reactivo en combinación con una metodología VNS (Variable Neighborhood Search) general; el segundo, basado en generación de columnas. La eficiencia computacional

de ambos procedimientos es probada sobre un gran conjunto de instancias de prueba.

## 1.1 PROBLEMAS DE RUTEO DE VEHÍCULOS CON BENEFICIOS

El OP pertenece a la clase de problemas de ruteo de vehículos con beneficios, (Vehicle Routing Problems with Profits, Archetti *et al.* (2013b); Traveling Salesman Problems with Profits, Feillet *et al.* (2005)). Contrario al caso de los clásicos problemas de ruteo de vehículos, en este tipo de problemas no se conoce a priori el conjunto de clientes a visitar, sino que se tiene un conjunto de clientes potenciales.

Cada uno de los posibles clientes tiene asociada una puntuación o beneficio que es colectado al servirlo y, debido a que la duración del recorrido no puede exceder un límite o a que el puntaje colectado no puede ser menor que una cota fija, se debe decidir a cuáles clientes visitar y en qué orden, simultáneamente.

Dependiendo del objetivo y de las restricciones del problema, en la literatura se distinguen varios tipos de problemas de ruteo de vehículos con beneficios, los cuales se presentan a continuación y se resumen en la Tabla 1.1.

Si solo se dispone de un vehículo para visitar a los clientes, se tienen los siguientes problemas:

- **Orienteering Problem (OP).** El objetivo es maximizar la puntuación colectada, sujeto a que la duración de la ruta no puede exceder un límite fijo. Fue introducido por Tsiligirides (1984) y también es conocido en la literatura como *Selective Traveling Salesman Problem* (Laporte y Martello, 1990), *Maximum Collection Problem* (Kataoka y Morito, 1988) y *Bank Robber Problem* (Arkin *et al.*, 1998).
- **Prize Collecting Traveling Salesman Problem (PCTSP).** Este problema fue estudiado por primera vez por Balas (1989) y el propósito es el de

encontrar la ruta cuya duración sea mínima y cuya puntuación colectada sea mayor o igual que un valor mínimo requerido.

- **Profitable Tour Problem (PTP)**. En este problema se combinan los objetivos de maximizar la puntuación y minimizar la duración del recorrido a través de la minimización de la diferencia entre la longitud del recorrido y el puntaje colectado. El PTP fue introducido en la literatura por Dell'Amico *et al.* (1995).

En el caso de que se cuente con múltiples vehículos, se distinguen los siguientes problemas:

- **Team Orienteering Problem (TOP)**. El TOP es una extensión del OP, se busca maximizar la puntuación colectada por todos los recorridos, sujeto a que existe un límite en la duración de cada uno de ellos. Dicha extensión fue realizada por primera vez por Butt y Cavalier (1994) bajo el nombre de *Multiple Tour Maximum Collection Problem* (MTMCP). Chao *et al.* (1996b) fueron quienes se refirieron a este problema con el nombre de *Team Orienteering Problem* por primera vez.
- **Capacitated Profitable Tour Problem (CPTP)**. El objetivo del CPTP es el de maximizar la diferencia entre el puntaje total colectado y el costo total de recorrido, este problema es una extensión del PTP considerando, además, que cada vehículo posee una capacidad limitada. El problema fue estudiado por primera vez por Archetti *et al.* (2009a).

<b>Problema</b>	<b>Número de rutas</b>	<b>Objetivo</b>	<b>Restricciones</b>
Orienteering Problem	Una	Maximizar la puntuación colectada	Restricciones de TSP y la duración de la ruta no debe exceder un límite fijo
Prize Collecting Traveling Salesman Problem	Una	Minimizar la duración de la ruta	Restricciones de TSP y el puntaje colectado es mayor o igual que un límite fijo
Profitable Tour Problem	Una	Minimizar la diferencia entre la duración de la ruta y el puntaje colectado	Restricciones de TSP
Team Orienteering Problem	Múltiples	Maximizar la puntuación colectada	Restricciones de VRP y la duración de las rutas no puede exceder un límite fijo
Capacitated Profitable Tour Problem	Múltiples	Minimizar la diferencia entre la duración total de las rutas y el puntaje colectado	Restricciones de VRP

Tabla 1.1: Problemas de ruteo de vehículos con beneficios

## 1.2 MOTIVACIÓN DE LA TESIS

El OP cuenta con múltiples aplicaciones documentadas en la literatura. Una de estas aplicaciones es la entrega de combustible a domicilio (Golden *et al.*, 1987), en este caso se tiene un conjunto de vehículos que deben servir a clientes cuya cantidad de combustible debe mantenerse en cierto nivel. La prioridad de cada cliente

está asociada al nivel de inventario y, diariamente, se elige al conjunto de clientes a servir.

Otra aplicación del OP es el diseño de recorridos turísticos (Souffriau *et al.*, 2008). En este caso, un turista que desea conocer una ciudad y que tiene presupuesto o tiempo limitado, asigna a cada punto de interés un beneficio por visitarlo.

Además de los trabajos mencionados anterioremente, en la práctica existen problemas de planificación que pueden ser vistos como un OP con restricciones adicionales. Por ejemplo, consideremos el problema referente a la planeación de rutas de recolección de residuos peligrosos. Para un día particular, a cada cliente se le asocia un beneficio proporcional a la urgencia del servicio y la duración máxima de la ruta corresponde a la jornada laboral de los empleados. Este problema puede ser abordado como un OP; no obstante, en este problema se debe tomar en cuenta que existen productos que no pueden ser transportados en el mismo vehículo, ya que pueden tener reacciones peligrosas que pondrían en riesgo a los empleados encargados de la recolección. Por otro lado, si la realización de un servicio no puede esperar al día siguiente, se debe visitar al cliente que lo solicita de manera obligatoria.

Por otra parte, si bien en la literatura se pueden encontrar distintas versiones de problemas de ruteo de vehículos con beneficios en los cuales se han integrado restricciones de visitas obligatorias y/o conflictos entre clientes, el problema propuesto en esta tesis no ha sido estudiado previamente. Debido a ello, se requiere un generador de instancias del problema y es importante desarrollar algoritmos de optimización eficientes para dar solución al mismo y a problemáticas reales que pueden ser modeladas como un OP con restricciones adicionales.

### 1.3 METODOLOGÍA DE INVESTIGACIÓN

La metodología que se siguió para el trabajo de tesis consistió en los siguientes pasos:

- Revisión del estado del arte de problemas de transporte y de ruteo vehículos clásicos y con beneficios relacionados con el problema planteado en la tesis.
- Formulación matemática del problema de estudio.
- Diseño de instancias del problema basadas en instancias existentes en la literatura del OP.
- Solución de las instancias generadas haciendo uso del solver Cplex 12.6, con la finalidad de detectar el alcance de los modelos matemáticos.
- Diseño e implementación computacional del algoritmo metaheurístico.
- Diseño e implementación computacional de un método de generación de columnas.
- Experimentación computacional sobre las instancias previamente diseñadas, con el propósito de determinar la calidad de los algoritmos propuestos.
- Análisis de los resultados obtenidos por ambos algoritmos.
- Escritura de la tesis.

## 1.4 ESTRUCTURA DE LA TESIS

La manera en que se estructura la tesis es la siguiente. En el Capítulo 2 se presenta una descripción formal del problema planteado, así como dos formulaciones matemáticas que abordan desde enfoques distintos la eliminación de subtours y se discute el estado del arte, para posteriormente exponer comparaciones entre los resultados obtenidos al resolver estos modelos en el Capítulo 3.

En el Capítulo 4, se describe el algoritmo heurístico, presentando las componentes de los esquemas GRASP y VNS y la manera en que estas metaheurísticas se adaptaron para resolver el problema planteado; los resultados obtenidos por dicho algoritmo se presentan en el Capítulo 5.

---

En el Capítulo 6, se presenta un modelo alternativo basado en la descomposición del problema en un problema maestro y un subproblema, para ser resuelto a través de un método de generación de columnas y, finalmente, en el Capítulo 7 se discuten las conclusiones de la tesis y el trabajo futuro.

## CAPÍTULO 2

# PLANTEAMIENTO DEL PROBLEMA

---

El OP es un problema que ha sido ampliamente estudiado en la literatura, desde que fue introducido se han propuesto diferentes métodos para su solución, tanto heurísticos como exactos y múltiples aplicaciones reales han sido estudiadas a través de él. No obstante, no se conoce trabajo alguno que haya estudiado la versión del problema que se propone en esta tesis, aunque existen propuestas que comparten ciertas similitudes con él, como lo son las visitas obligatorias y los conflictos.

En este capítulo se describe el estado del arte del OP, asimismo se detallan las características de los trabajos existentes en la literatura que presentan similitudes con el problema de estudio. Posteriormente, se describe el OP con restricciones de conflictos y visitas obligatorias de manera formal y se presentan dos formulaciones matemáticas que se proponen para su modelación.

## 2.1 ESTADO DEL ARTE DEL PROBLEMA DEL AGENTE VIAJERO SELECTIVO

El Orienteering Problem está inspirado en el deporte *Orienteering* en su variante *Score*, en la cual los competidores utilizan un mapa y una brújula para visitar ciertos puntos de control dentro de un tiempo límite, dichos puntos tienen diferentes puntuaciones de acuerdo a la dificultad que representan. Si el competidor excede el tiempo límite, es descalificado y gana la competencia aquel jugador que obtenga la

mayor puntuación.

El OP fue introducido por Tsiligirides (1984) y en la literatura ha sido estudiado bajo distintos nombres, como lo son *Selective Traveling Salesman Problem* (Laporte y Martello, 1990), *Maximum Collection Problem* (Kataoka y Morito, 1988) y *Bank Robber Problem* (Arkin *et al.*, 1998).

Como se mencionó en el capítulo anterior, una extensión del OP se obtiene al disponer de múltiples vehículos para visitar a los clientes. Dicha extensión es conocida como *Team Orienteering Problem* (Butt y Cavalier, 1994; Chao *et al.*, 1996b). En el TOP se genera un conjunto de rutas con el objetivo de maximizar la puntuación colectada por las mismas, tomando en cuenta que la duración de cada una de ellas no debe exceder el límite prefijado. Al incluirse ventanas de tiempo a los nodos, el OP y el TOP se pueden extender al *Orienteering Problem with Time Windows* (OPTW) (Righini y Salani, 2006a) y al *Team Orienteering Problem with Time Windows* (TOPTW) (Vansteenwegen, 2009), respectivamente.

En el *Time Dependent Orienteering Problem* (TDOP) (Fomin y Lingas, 2002), el tiempo de traslado de un sitio a otro no es fijo, sino que depende de la hora de partida y el *Time Dependent Orienteering Problem with Time Windows* (TDOPTW), Garcia *et al.* (2010) generaliza al TDOP y al OPTW.

Al tomar en cuenta restricciones de demanda de cada cliente y de capacidad en cada vehículo, el TOP se extiende al *Capacitated Team Orienteering Problem* (CTOP), propuesto por Archetti *et al.* (2009a).

Una versión multiobjetivo del OP fue introducida por Schilde *et al.* (2009), en la cual existen distintas categorías de evaluación y cada nodo posee una puntuación distinta para cada una de ellas; entonces, el interés recae en maximizar la puntuación colectada en cada clasificación.

Múltiples aplicaciones del OP y sus variantes pueden ser encontradas en la literatura, como lo son la entrega de combustible a domicilio (Golden *et al.*, 1987), el reclutamiento de atletas en escuelas (Butt y Cavalier, 1994), el diseño del horario

de técnicos que sirven a clientes en distintas localizaciones geográficas (Tang y Miller-Hooks, 2005) y la planeación de recorridos turísticos (Vansteenwegen *et al.*, 2009; Garcia *et al.*, 2010; Vansteenwegen *et al.*, 2011a).

En cuanto a su complejidad computacional, está demostrado que el OP es NP-duro (Golden *et al.*, 1987), por lo cual se han propuesto múltiples algoritmos de tipo heurístico para su solución (Tsiligirides, 1984; Golden *et al.*, 1987; Ramesh y Brown, 1991; Butt y Cavalier, 1994; Chao *et al.*, 1996a).

Del mismo modo, también se han propuesto procedimientos metaheurísticos como lo son redes neuronales artificiales (Wang *et al.*, 1995), algoritmos genéticos (Tasgetiren y Smith, 2000; Karbowska-Chilinska y Zabielski, 2013), algoritmos de colonia de hormigas (Liang *et al.*, 2002; Schilde *et al.*, 2009), búsqueda tabú (Gendreau *et al.*, 1998b; Liang *et al.*, 2002), búsqueda por entornos variables (Sevкли y Sevilgen, 2006; Schilde *et al.*, 2009), algoritmos de cúmulo de partículas (Sevкли y Sevilgen, 2010) y procedimiento de búsqueda miope aleatorizado y adaptativo con reencadenamiento de trayectorias (GRASP con path relinking) (Campos *et al.*, 2014).

Por otro lado, también existen propuestas de solución de tipo exacto como lo son programación dinámica (Hayes y Norman, 1984), *branch and bound* (Kataoka y Morito, 1988; Laporte y Martello, 1990; Ramesh *et al.*), planos cortantes (Leifer y Rosenwein, 1994), y *branch and cut* (Fischetti *et al.*, 1998; Gendreau *et al.*, 1998a).

El método propuesto por Fischetti *et al.* (1998) es capaz de resolver instancias de hasta 500 nodos, siendo el algoritmo de tipo exacto que ha logrado resolver instancias de mayor tamaño. En cuanto a los métodos heurísticos, el GRASP con reencadenamiento de trayectorias propuesto por Campos *et al.* (2014) obtuvo los mejores resultados conocidos hasta el momento en las instancias de Chao *et al.* (1996a), además de haber obtenido una desviación con respecto al óptimo de 1.4% en promedio en las instancias de la clase 1 de Fischetti *et al.* (1998). Sin embargo, el algoritmo genético de Karbowska-Chilinska y Zabielski (2013) obtuvo mejores

resultados que el GRASP en un conjunto de instancias cuyo tamaño es de 908 nodos.

## 2.2 ESTADO DEL ARTE DEL PROBLEMA DEL AGENTE VIAJERO SELECTIVO CON UNO O MÚLTIPLES AGENTES Y RESTRICCIONES DE VISITAS OBLIGATORIAS Y CONFLICTOS ENTRE NODOS

A pesar de que la literatura del OP es vasta, no se logró encontrar un trabajo que haya estudiado la versión que se propone en esta tesis. No obstante, se identificaron algunos artículos cuyas propuestas comparten ciertas similitudes con este, los cuales se resumen en la Tabla 2.1.

La existencia de nodos que deben visitarse de manera obligatoria en un OP fue abordada por primera vez por Gendreau *et al.* (1998a). Mediante el uso de un algoritmo *branch and cut*, lograron resolver a optimalidad instancias desde 20 hasta 300 nodos con 25 y 50% de nodos obligatorios.

Una extensión del TOPTW llamada *Multi-Period Orienteering Problem with Multiple Time Windows* (MPOPTW), fue estudiada por Tricoire *et al.* (2010) para el desarrollo de un sistema de planificación de visitas de vendedores a clientes, los cuales poseen múltiples ventanas de tiempo que pueden ser diferentes en distintos días. Las visitas a clientes con los cuales se tiene una relación de largo plazo son consideradas obligatorias y deben hacerse en periodos regulares. Por otro lado, las visitas opcionales se asocian a clientes potenciales y su beneficio guarda relación con el producto de la probabilidad de que este se convierta en cliente por la ganancia recibida. Para la solución del MPOPTW, se propuso algoritmo basado en un VNS en combinación con un procedimiento exacto que resuelve el problema de factibilidad de la ruta.

Por otro lado, Vansteenwegen *et al.* (2011a) implementan un sistema de nave-

gación turística, para el cual tienen una base de datos que contiene los puntos de interés de cierta ciudad, junto con sus coordenadas, sus horarios de apertura y cierre y el promedio de la duración de la visita. La oficina de turismo local marca ciertos puntos como *'not to be missed'*, por lo cual, estos son incluidos en el recorrido y el resto de los lugares a visitar es elegido de manera que se maximice la puntuación estimada en base a los intereses de los usuarios. Este problema es una extensión del TOPTW en la cual cada sitio puede tener múltiples ventanas de tiempo que pueden ser diferentes en distintos días y, además, se añade la posibilidad de incluir un descanso para comida; el método de solución empleado para resolverlo está basado en un esquema GRASP.

El estudio de conflictos entre nodos y visitas obligatorias en un problema de tipo OP fue abordado por primera vez por Salazar-Aguilar *et al.* (2014) en un *Multi-District Team Orienteering Problem* (MDTOP). El MDTOP fue utilizado para desarrollar la planeación de las actividades de flotillas de mantenimiento vial. Se considera un horizonte de planeación que es dividido entre múltiples distritos, en cada uno de ellos existen actividades que son vitales para mantener el correcto funcionamiento del flujo vehicular y otras actividades que pueden esperar al siguiente horizonte de planeación. Existen actividades que no pueden realizarse en un mismo día, ya que impedirían el tránsito como, por ejemplo, realizar reparaciones en ambos sentidos de la misma calle. Para la solución del MDTOP se propuso un algoritmo tipo ALNS.

En esta tesis se considera un OP con visitas obligatorias y con conflictos entre nodos, haciendo uso de un procedimiento basado en un GRASP reactivo cuya fase de postprocesamiento sigue un esquema VNS general. Por otro lado, cotas para el problema propuesto, son obtenidas mediante un algoritmo de generación de columnas.

<b>Autores</b>	<b>Número de rutas</b>	<b>Múltiples distritos</b>	<b>Ventanas de tiempo</b>	<b>Visitas obligatorias</b>	<b>Conflictos</b>	<b>Método de solución</b>
Gendreau <i>et al.</i> (1998a)	Una	No	No	Sí	No	Branch and cut
Tricoire <i>et al.</i> (2010)	Múltiples	No	Sí, múltiples	Sí	No	VNS
Vansteenwegen <i>et al.</i> (2011a)	Múltiples	No	Sí, múltiples	Sí	No	GRASP
Salazar-Aguilar <i>et al.</i> (2014)	Múltiples	Sí	No	Sí	Sí	ALNS
Palomo	Una	No	No	Sí	Sí	GRASP con VNS y Generación de columnas

Tabla 2.1: Estado del arte del OP y el TOP con visitas obligatorias y conflictos

## 2.3 DESCRIPCIÓN FORMAL DEL PROBLEMA

Dado un conjunto de clientes para los cuales se conoce su localización, el problema planteado consiste diseñar una ruta que comience y termine en un depósito (o depósitos distintos), que visite a todos los clientes obligatorios y a algunos opcionales con el objetivo de maximizar el beneficio colectado al servir a estos últimos, tomando en cuenta que la duración del recorrido no puede exceder un límite dado y que existen conflictos entre clientes.

De manera formal, considere el grafo completo no dirigido  $G = (V, E)$  en el cual se asume que se cumple la desigualdad triangular. Sean  $M \subset V$  el conjunto de nodos que deben incluirse en la ruta de manera obligatoria y  $O \subset V$  el conjunto de nodos opcionales, para los cuales se cumple que  $M \cap O = \emptyset$  y  $M \cup O = V \setminus \{1, n\}$ , donde  $n = |V|$ . Para cada nodo  $i \in O$  se conoce  $s_i$ , el beneficio recibido por visitarlo; también para cada nodo  $i \in V \setminus \{1, n\}$  se conoce el conjunto  $C_i$  de nodos con los

cuales este tiene conflicto. Además para todo arco  $(i, j) \in E$  se conoce el tiempo de recorrido, el cual es denotado por  $t_{ij}$ .

Una ruta factible  $r$ , debe comenzar en el nodo 1, terminar en el  $n$ , debe incluir a todos los nodos obligatorios, no debe incluir nodos que tengan conflicto y su duración  $d(r)$  no debe exceder un límite máximo denotado por  $t_{max}$ .

En cuanto a los nodos obligatorios se hace la siguiente observación:

**Observación 2.1** *Los nodos obligatorios tienen un beneficio asociado igual a cero, ya que si se les asignase una puntuación positiva, esta sería una constante en la función objetivo.*

## 2.4 FORMULACIÓN MATEMÁTICA DEL PROBLEMA DEL AGENTE VIAJERO SELECTIVO CON RESTRICCIONES ADICIONALES

### 2.4.1 MODELO MATEMÁTICO

En esta tesis se proponen dos formulaciones matemáticas de programación lineal entera mixta. Considerando la notación de la sección anterior y las variables de decisión  $x_{ij} = 1$  si el nodo  $j$  es visitado inmediatamente después del nodo  $i$  y  $x_{ij} = 0$  en otro caso, y  $y_i = 1$  si el nodo  $i \in V$  es visitado y  $y_i = 0$  si no; el modelo propuesto para el problema planteado es descrito por las expresiones (2.1) a (2.10).

$$\max \sum_{i \in O} s_i y_i \quad (2.1)$$

sujeto a:

$$\sum_{j \in V \setminus \{1\}} x_{1j} = 1 \quad (2.2)$$

$$\sum_{i \in V \setminus \{n\}} x_{in} = 1 \quad (2.3)$$

$$y_k = 1 \quad k \in M \quad (2.4)$$

$$\sum_{j \in V \setminus \{n\}, i \neq j} x_{ji} = y_i \quad i \in V \setminus \{1\} \quad (2.5)$$

$$\sum_{j \in V \setminus \{1\}, i \neq j} x_{ij} = y_i \quad i \in V \setminus \{n\} \quad (2.6)$$

$$\sum_{j \in C_i} y_j + y_i \leq 1 \quad i \in V \setminus \{1, n\}, C_i \neq \emptyset \quad (2.7)$$

$$\sum_{i \in V \setminus \{n\}} \sum_{j \in V \setminus \{1\}} t_{ij} x_{ij} \leq t_{max} \quad (2.8)$$

$$x_{ij} \in \{0, 1\} \quad i, j = 1, 2, \dots, n \quad (2.9)$$

$$y_i \in \{0, 1\} \quad i = 1, 2, \dots, n \quad (2.10)$$

La expresión (2.1) busca maximizar la puntuación colectada al visitar a los nodos opcionales que conforman la ruta, las restricciones (2.2) y (2.3) aseguran que la ruta comienza en el nodo 1 y termina en el  $n$ . El grupo de restricciones (2.4) asegura que todos los nodos obligatorios serán visitados. Por otro lado, las restricciones (2.5) y (2.6) corresponden a la conservación de flujo en el nodo  $i$ . El conjunto de restricciones (2.7) aseguran que la solución no incluya simultáneamente a nodos con conflicto entre ellos. La restricción (2.8) asegura que la longitud de la ruta no exceda el límite y, por último, las restricciones (2.9) y (2.10) determinan la naturaleza de las variables.

Si los conflictos entre nodos cumplen las propiedades simétrica y transitiva, es decir, si  $\forall i, j, k \in V \setminus \{1, n\}$ , se cumple que:

1. Si  $i$  tiene conflicto con  $j$ , entonces  $j$  tiene conflicto con  $i$ ,
2. Si  $i$  tiene conflicto con  $j$  y  $j$  tiene conflicto con  $k$ , entonces  $i$  tiene conflicto con  $k$ ,

entonces el número de restricciones (2.7) puede ser reducido debido a la siguiente proposición.

**Proposición 2.2** *Consideremos la relación binaria  $\mathfrak{R}$  definida sobre  $V \setminus \{1, n\}$ , decimos que  $i\mathfrak{R}j$  si y solo si  $i$  tiene conflicto con  $j$ , i.e.  $j \in C_i$ . Si  $\mathfrak{R}$  es simétrica y transitiva, entonces  $\sum_{k \in C_i} y_k + y_i = \sum_{l \in C_j} y_l + y_j$ .*

**Demostración:** Sean  $i, j \in V \setminus \{1, n\}$  tales que  $i\mathfrak{R}j$ .

1. Sea  $x \in C_i \cup \{i\}$ .

- a) Si  $x = i$ , entonces  $x\mathfrak{R}j$ . Ahora bien, por simetría de  $\mathfrak{R}$ , se tiene que  $j\mathfrak{R}x$ . En consecuencia, por la definición de la relación binaria, se sigue que  $x \in C_j$ .

$$\therefore x \in C_j \cup \{j\} \quad (2.11)$$

- b) Si  $x \in C_i$ , por definición de  $\mathfrak{R}$  se tiene que  $i\mathfrak{R}x$ . Ahora, como  $i\mathfrak{R}j$ , entonces  $j\mathfrak{R}i$ , debido a la simetría de la relación binaria. De lo anterior, por transitividad se tiene que  $j\mathfrak{R}x$  y, por definición de  $\mathfrak{R}$ , se sigue que  $x \in C_j$ .

$$\therefore x \in C_j \cup \{j\} \quad (2.12)$$

Por lo tanto, de (2.11) y (2.12) se concluye que

$$C_i \cup \{i\} \subset C_j \cup \{j\} \quad (2.13)$$

2. Sea  $x \in C_j \cup \{j\}$

a) Si  $x = j \Rightarrow x \in C_i$ , dado que  $i\mathfrak{R}j$ .

$$\therefore x \in C_i \cup \{i\} \quad (2.14)$$

b) Si  $x \in C_j$ , entonces  $j\mathfrak{R}x$ , pero como  $i\mathfrak{R}j$ , por la propiedad transitiva de  $\mathfrak{R}$  se tiene que  $i\mathfrak{R}x$ . Como consecuencia de la definición de la relación binaria,  $x \in C_i$ .

$$\therefore x \in C_i \cup \{i\} \quad (2.15)$$

Por lo tanto, de las expresiones (2.14) y (2.15) se concluye que

$$C_i \cup \{i\} \supset C_j \cup \{j\} \quad (2.16)$$

Como consecuencia de (2.13) y (2.16) se tiene que  $C_i \cup \{i\} = C_j \cup \{j\}$ . Entonces, se tiene que:

$$\sum_{k \in C_i \cup \{i\}} y_k = \sum_{l \in C_j \cup \{j\}} y_l \quad (2.17)$$

$$\Rightarrow \sum_{k \in C_i} y_k + y_i = \sum_{l \in C_j} y_l + y_j \quad (2.18)$$

Por lo tanto, si la relación  $\mathfrak{R}$  es simétrica y transitiva y si  $i\mathfrak{R}j$ , i.e.  $j \in C_i$ , se cumple que:

$$\sum_{k \in C_i} y_k + y_i = \sum_{l \in C_j} y_l + y_j \quad (2.19)$$

■

En esta tesis se asume que los conflictos cumplen la Proposición 2.2, por lo

cual los nodos que presentan incompatibilidades pueden ser agrupados en conjuntos disjuntos denotados por  $C'_1, C'_2, \dots, C'_L$  tales que si  $i, j \in C'_k$  con  $1 \leq k \leq L$ , entonces  $i$  y  $j$  son incompatibles. Como consecuencia, el grupo de restricciones (2.7) se reescribe como:

$$\sum_{j \in C'_i} y_j \leq 1, \quad i = 1, 2, \dots, L \quad (2.20)$$

Nótese que al modelo descrito anteriormente se le deben añadir restricciones de eliminación de ciclos; la manera en que se aborda el estudio de dichas restricciones se discutirá en la siguiente sección.

## 2.4.2 RESTRICCIONES DE ELIMINACIÓN DE CICLOS PARA EL OP

En la literatura de problemas de ruteo de vehículos es común que se utilicen restricciones de eliminación de ciclos inspiradas en las propuestas por Miller *et al.* (1960) para el TSP, debido a la facilidad de su implementación y a que son capaces de resolver instancias de tamaño mediano en tiempos de cómputo razonables (Bektaş y Gouveia, 2014). Sin embargo, poseen la desventaja de proveer una relajación lineal débil (Langevin *et al.* (1990); Gouveia y Pires (1999)).

Por otro lado, el conjunto de restricciones propuesto por Dantzig *et al.* (1954) produce una relajación lineal fuerte, pero su uso no es popular debido a que contiene un número exponencial de restricciones.

Debido a lo anterior, múltiples propuestas para evitar la formación de ciclos pueden ser encontradas en la literatura del TSP (Vajda, 1961; Gavish y Graves, 1978; Fox *et al.*, 1980; Wong, 1980; Finke *et al.*; Claus, 1984).

Para este trabajo, se adaptaron dos grupos de restricciones existentes en la literatura del TSP: el primero está conformado por restricciones secuenciales inspiradas en las propuestas por Miller *et al.* (1960); el segundo está compuesto por

restricciones basadas en flujo de múltiples productos siguiendo el esquema de las propuestas por Wong (1980) y Claus (1984). Estas últimas fueron elegidas debido a que Wong (1980) demostró que el politopo producido por la relajación lineal del modelo del TSP que utiliza estas restricciones es igual al politopo resultante de la relajación lineal de la formulación que hace uso de las restricciones propuestas por Dantzig *et al.* (1954).

#### RESTRICCIONES DE ELIMINACIÓN DE CICLOS SECUENCIALES

Sea  $u_i$  la variable entera que representa la posición en la ruta del nodo  $i$ , las restricciones de eliminación de ciclos para el TSP propuestas por Miller *et al.* (1960) están representadas por la expresiones siguientes.

$$2 \leq u_i \leq n \quad i = 2, 3, \dots, n \quad (2.21)$$

$$u_i - u_j + 1 \leq (n - 1)(1 - x_{ij}) \quad \forall i, j \in V \quad (2.22)$$

En el caso del OP, las restricciones anteriores han sido utilizadas para prevenir la formación de ciclos; sin embargo, dado que no todos los nodos del grafo deben ser visitados, las variables  $u_i$  no necesariamente representan la posición del nodo  $i$  en la ruta. Para retomar esta definición, basta con acotar las variables  $u_i$ , con lo cual la expresión (2.21) se reformula como en la expresión (2.23)

$$2 \leq u_i \leq \sum_{j \in V} y_j \quad i = 2, 3, \dots, n \quad (2.23)$$

En conclusión, el modelo propuesto para el OP con restricciones de visitas obligatorias y conflictos entre clientes está conformado por las expresiones (2.1) a (2.6), (2.8) a (2.10), (2.20), (2.22) y (2.23).

## RESTRICCIONES DE ELIMINACIÓN DE CICLOS BASADAS EN FLUJO

Considere la variable continua no negativa  $z_{ij}^k$  que representa el flujo del producto  $k$  en el arco  $(i, j)$ , para  $k = 1, 2, \dots, n$  y  $(i, j) \in E$ . Las expresiones (2.24) a (2.29) representan las restricciones propuestas por Wong (1980) y Claus (1984) para eliminar ciclos.

$$z_{ij}^k \leq x_{ij} \quad i, j \in V, k = 1, 2, \dots, n \quad (2.24)$$

$$\sum_{i \in V} z_{1i}^k = 1 \quad k = 2, 3, \dots, n \quad (2.25)$$

$$\sum_{i \in V} z_{i1}^k = 0 \quad k = 2, 3, \dots, n \quad (2.26)$$

$$\sum_{i \in V} z_{ik}^k = 1 \quad k = 2, 3, \dots, n \quad (2.27)$$

$$\sum_{i \in V} z_{ki}^k = 0 \quad k = 2, 3, \dots, n \quad (2.28)$$

$$\sum_{i \in V} z_{ij}^k - \sum_{i \in V} z_{ji}^k = 0 \quad k = 2, 3, \dots, n, j \in V \setminus \{1\}, j \neq k \quad (2.29)$$

Las restricciones (2.24) solo permiten el flujo en un arco si este es utilizado en la ruta. Las restricciones (2.25) aseguran que el nodo 1 sea la fuente de una unidad de flujo de cada producto y las restricciones (2.26) evitan que el flujo regrese al nodo 1. Las restricciones (2.27) y (2.28) fuerzan a que una unidad de flujo del producto  $k$  entre al nodo  $k$  y que no salga de él. Las restricciones (2.29) aseguran la conservación de flujo.

Debido a que en el OP una solución factible es un camino y no un ciclo, se adecuaron las restricciones (2.26) y (2.27); además, hay que hacer la consideración de que no todos los nodos son visitados. La adaptación resultante está dada por las restricciones (2.30) y (2.31), respectivamente.

$$\sum_{i \in V} z_{in}^k = 1 - y_k \quad k = 2, 3, \dots, n - 1 \quad (2.30)$$

$$\sum_{i \in V} z_{ik}^k = y_k \quad k = 2, 3, \dots, n - 1 \quad (2.31)$$

Las restricciones (2.30) aseguran que si un nodo no fue visitado, una unidad de flujo del producto asociado con este llegará al nodo  $n$ ; si fue visitado, no llegará flujo de dicho producto. Las restricciones (2.31) fuerzan a que si un nodo es visitado, una unidad de flujo de su producto asociado debe entrar a él.

Por otro lado, cabe hacer la observación de que no se asocia un producto al nodo  $n$  porque, en el caso de hacerlo, las restricciones (2.30) y (2.31) serían contradictorias para  $k = n$ .

Finalmente, la formulación del problema planteado en esta tesis, cuya eliminación de ciclos se basa en restricciones de flujo, está dada por las expresiones (2.1) a (2.6), (2.8) a (2.10), (2.20), (2.24), (2.25) y (2.28) a (2.31).

## CAPÍTULO 3

# COMPARACIONES ENTRE LOS MODELOS PROPUESTOS

---

En la literatura del TSP se sabe que la formulación cuyas restricciones de eliminación de ciclos se basan en flujo de múltiples productos provee una relajación lineal más fuerte que aquella cuya eliminación de ciclos es abordada con restricciones de tipo secuencial. Sin embargo, también se sabe el modelo con restricciones de flujo tiene un mayor número de variables y de restricciones.

En este capítulo se presenta un análisis del número de restricciones y variables de los modelos propuestos en el capítulo anterior. Asimismo, se muestran las soluciones obtenidas al resolver ambas formulaciones mediante el uso de Cplex 12.6, en un gran conjunto de instancias de prueba y, del mismo modo, se exponen los valores obtenidos por sus respectivas relajaciones lineales.

### 3.1 DIMENSIÓN DE LOS MODELOS PROPUESTOS

En adelante, a la formulación matemática cuyas restricciones de eliminación de ciclos están basadas en flujo, se le denotará como M1; y el modelo en el cual se utilizan restricciones de tipo secuencial, estará denotado por M2.

Consideremos la siguiente notación:

- $n$ : Número de nodos

- $m$ : Número de nodos obligatorios
- $c$ : Número de grupos de conflicto

La Tabla 3.1 muestra una comparativa entre la cantidad y el tipo de variables, así como del número de restricciones utilizadas en cada modelo.

Modelo	Número de restricciones	Número de variables enteras	Número de variables continuas
M1	$n(n^2 + 2) + m + c - 3$	$n(n + 1)$	$n^3$
M2	$n(n + 3) + m + c$	$n(n + 2) - 1$	0

Tabla 3.1: Comparación entre las dimensiones de los modelos

Cabe destacar que la diferencia entre el número de restricciones de M1 y la cantidad de restricciones de la formulación M2 es de  $n(n^2 - n - 1) - 3$ . Considere la función  $f : n \mapsto n(n^2 - n - 1) - 3$ , cuyo dominio se define por  $\mathbb{R}$ , dicha función polinomial posee una raíz real  $n_0 \approx 2.1304$  y dos raíces complejas, además de ser creciente en el intervalo  $[n_0, +\infty)$ , tal como se muestra en la Figura 3.1.

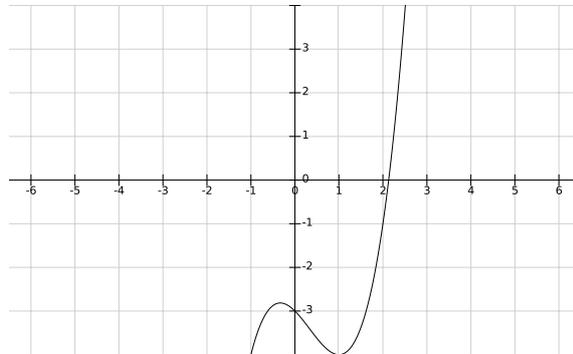


Figura 3.1: Gráfica de la diferencia entre el número de restricciones de los modelos

En conclusión de lo anterior, el número de restricciones de la formulación M1 será mayor que la cantidad de restricciones de la M2 si el número de nodos de la instancia es mayor que 2, es decir, en cualquier caso.

Por otro lado, el número de variables enteras utilizadas por la formulación M2 es, en todo caso, mayor que el número de variables enteras presentes en M1, ya que

$n(n+1) - 1 + [n(n+1)] = n - 1$  que es mayor que cero siempre que  $n > 1$ .

Por último en cuanto a la cantidad total de variables tanto enteras como continuas, el número en M1 es mayor que el de M2 para cualquier grafo. Para lo anterior, denotemos como  $g$  a la función cuyo dominio comprende al conjunto de los números reales y que está definida como la diferencia entre el número de variables entre M1 y M2, i.e.  $g : n \mapsto n^3 - n + 1$ .

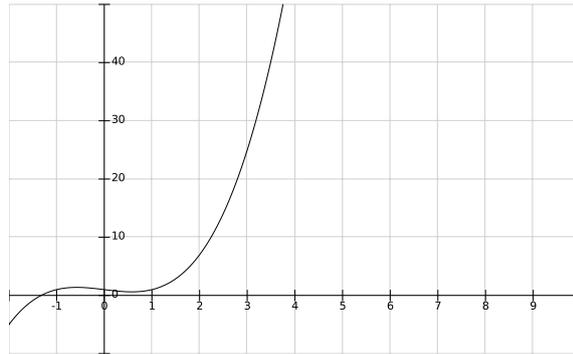


Figura 3.2: Gráfica de la diferencia entre el número de variables de los modelos

Como se muestra en la Figura 3.2,  $g$  tiene una raíz real  $n_0 \approx -1.3247$  y dos raíces complejas. Además, la función es creciente en el intervalo  $[n_0, +\infty)$ . Por lo tanto, el número total de variables utilizadas en M1 es mayor que la cantidad de variables usadas por M2 en todo caso.

## 3.2 ENTORNO EXPERIMENTAL

Se realizaron pruebas computacionales con el propósito de detectar el alcance de los modelos propuestos y de comprobar la calidad de la relajación lineal que estos proveen.

La experimentación computacional se realizó en una Workstation HP Z620 con procesador Intel Xeon(R) CPU E5-2620 v2 a 2.10 GHz y con memoria RAM de 64 Gb, bajo el sistema operativo Ubuntu 14.04 LTS. Los modelos fueron codificados en GAMS y resueltos haciendo uso del solver Cplex 12.6.

### 3.3 DESCRIPCIÓN DE LAS INSTANCIAS DE PRUEBA

Para realizar experimentación computacional se generaron seis grupos de instancias aleatorias. El primer grupo y el cuarto contienen 62 instancias; el segundo y el quinto, 55; y el tercero y el sexto, 53. Las instancias de prueba están basadas en la clase 1 de las que fueron propuestas por Fischetti *et al.* (1998) para el OP.

Cada uno de los grupos está basado en el mismo conjunto de grafos, cuyas dimensiones varían desde 21 hasta 262 nodos. Entre cada conjunto varía el porcentaje de nodos obligatorios y el nivel de conflictos entre clientes. Los nodos obligatorios, al igual que los conflictos, se generaron de manera aleatoria y cada nodo puede estar libre de conflictos o ser incompatible con un mínimo de un nodo y un máximo de tres.

En la Tabla 3.2 se muestran las características de cada clase de instancias; asimismo, una descripción detallada de cómo estas fueron generadas puede ser encontrada en el Apéndice B.

Clase	1	2	3	4	5	6
<b>Nodos obligatorios (porcentaje)</b>	10	20	30	10	20	30
<b>Nivel de conflictos</b>	Alto	Alto	Alto	Bajo	Bajo	Bajo
<b>Número de instancias</b>	62	55	53	62	55	53

Tabla 3.2: Descripción de las características de los conjuntos de instancias de prueba

## 3.4 SOLUCIONES OBTENIDAS POR LOS MODELOS PROPUESTOS

Como se mencionó anteriormente, se resolvieron las instancias de prueba, con las dos formulaciones propuestas, mediante el uso del solver Cplex 12.6 y del sistema de modelación algebraica GAMS. En ambos casos, el tiempo máximo para la solución del modelo por instancia fue fijado en una hora.

En la Tabla 3.3 se muestra el número de instancias que fueron resueltas a optimalidad dentro del tiempo límite haciendo uso de cada modelo. Con el fin de hacer comparaciones consistentes, se muestra el tiempo promedio de solución de las instancias de cada clase que se lograron resolver a optimalidad por ambas formulaciones. Asimismo, se presenta la cardinalidad de la instancia con mayor cantidad de nodos que pudo ser resuelta en cada clase. En el último renglón, se muestran los mismos datos para el total de las instancias contenidas en las seis clases. En las Tablas A.1 a A.6 del Apéndice A se exponen los resultados obtenidos por las formulaciones propuestas en cada una de las instancias.

Clase	Modelo M1			Modelo M2		
	Número de instancias resueltas	Tiempo promedio (segundos)	Máximo tamaño de instancias resueltas	Número de instancias resueltas	Tiempo promedio (segundos)	Máximo tamaño de instancias resueltas
1	36/62	44.37	76 nodos	34/62	443.02	151 nodos
2	33/55	155.02	76 nodos	29/55	122.36	101 nodos
3	33/53	147.68	76 nodos	31/53	124.47	101 nodos
4	36/62	182.24	76 nodos	25/62	274.78	76 nodos
5	28/55	43.6	51 nodos	21/55	212.23	51 nodos
6	30/53	41.09	76 nodos	17/53	643.37	33 nodos
<b>Total</b>	196/340	109.01	76 nodos	157/340	282.1	151 nodos

Tabla 3.3: Comparaciones entre las instancias resueltas por las formulaciones propuestas

Cabe destacar que, si bien el modelo M2 puede resolver instancias con mayor cardinalidad, la formulación M1 es capaz de encontrar soluciones óptimas para una mayor cantidad de casos y en un tiempo promedio menor.

### 3.5 RELAJACIÓN LINEAL DE LOS MODELOS

#### PROPUESTOS

De acuerdo con la teoría del TSP, la formulación cuya eliminación de ciclos se basa en restricciones de flujo de múltiples productos provee una relajación lineal más fuerte que aquella con restricciones de eliminación de ciclos de tipo secuencial. En esta sección se muestran los resultados obtenidos al resolver la relajación lineal de los modelos M1 y M2, con el propósito de determinar de manera experimental si en el OP se cumple el mismo resultado.

Para cada uno de los modelos M1 y M2, se relajaron las restricciones de integralidad y se trató de resolver cada una de las instancias utilizando los nuevos modelos relajados MR1 y MR2, respectivamente. Para ello, se utilizó el solver Cplex 12.6, tomando como máximo una hora para el tiempo de solución. En la Tabla 3.4 se muestran los valores promedio de las soluciones obtenidas al resolver las relajaciones lineales de los modelos y el tiempo promedio de solución en cada clase, asimismo, en el último renglón se presentan los mismos datos para el total de las instancias. De manera similar a la sección anterior, para conservar la consistencia, dichos valores fueron obtenidos sobre las instancias que lograron ser resueltas dentro de una hora por ambas relajaciones.

Clase	Modelo relajado MR1		Modelo relajado MR2	
	Valor promedio de la relajación lineal	Tiempo promedio (segundos)	Valor promedio de la relajación lineal	Tiempo promedio (segundos)
1	2132.9221	277.04	2709.51345	0.04
2	1419.12698	211.09	1540.74709	0.03
3	804.769403	96.83	844.03125	0.02
4	2890.76736	125.76	4153.22196	0.03
5	2320.94394	191.16	2903.67	0.04
6	1730.70078	335.64	2033.33287	0.04
<b>Total</b>	1918.66414	243.47	2416.1006	0.03

Tabla 3.4: Comparaciones entre las instancias resueltas por las relajaciones lineales de las formulaciones propuestas

De la Tabla 3.4 se puede concluir que, aunque el tiempo necesario para obtener el valor de la cota superior provista por la relajación lineal del modelo M1 es mucho mayor, la calidad de dicha cota es superior, observándose un gap del 20.59%.

En la Figura 3.3 se puede observar el comportamiento de los valores de la relajación lineal obtenidos por MR1 y MR2 en las instancias que fueron resueltas por ambas relajaciones en comparación con los valores óptimos. En el eje horizontal se representan las instancias y en el vertical, el valor de la cota superior obtenida por las respectivas relajaciones y el valor óptimo en una escala logarítmica.

Por último, cabe destacar que haciendo uso de la relajación MR2 se logró encontrar una cota superior para el valor óptimo de todas las instancias; sin embargo, con la relajación MR1 solamente lograron resolver 220 instancias, es decir, un 64.7% del total. Aun así, se consiguió mostrar de manera experimental que, en efecto, un modelo con restricciones de eliminación de ciclos basadas en flujo provee una relajación lineal más fuerte para el OP que la formulación con restricciones de eliminación de ciclos de tipo secuencial.

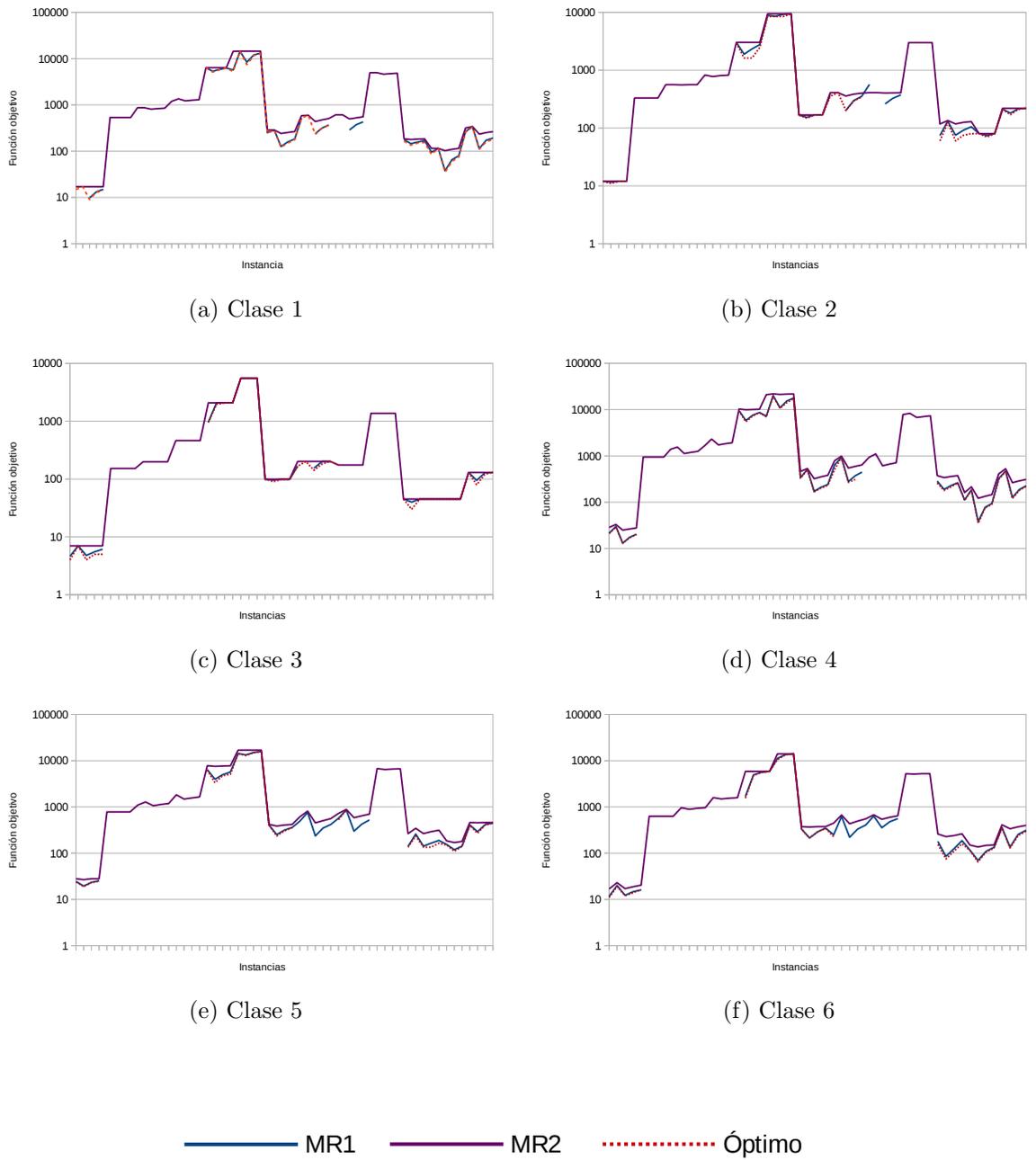


Figura 3.3: Comportamiento de las relaciones lineales de los modelos propuestos

## CAPÍTULO 4

# ALGORITMO HEURÍSTICO PROPUESTO

---

Greedy Randomized Adaptive Search Procedure, GRASP (Feo y Resende, 1989, 1995) es una metaheurística multiarranque, cuya versión más simple está compuesta por una etapa de construcción de una solución seguida de un postprocesamiento de la misma, usualmente, haciendo uso de un algoritmo de búsqueda local. El proceso continúa iterativamente hasta satisfacer un criterio de paro y se reporta la mejor solución obtenida.

Por otro lado el Variable Neighborhood Search, VNS (Mladenović y Hansen, 1997), es un esquema metaheurístico que se basa en la exploración del espacio de soluciones a través de búsqueda local haciendo uso de múltiples vecindarios.

En el estado del arte del OP, los esquemas GRASP y VNS han mostrado ser los más competitivos, arrojando las mejores soluciones conocidas hasta el momento (Schilde *et al.*, 2009; Campos *et al.*, 2014). Además, un esquema reactivo se ajusta automáticamente al comportamiento del algoritmo y disminuye la cantidad de parámetros que deben ser calibrados. Debido a lo anterior, en esta tesis se propone un algoritmo tipo GRASP reactivo cuya fase de postprocesamiento es un esquema basado en un VNS general.

En este capítulo, se describirán las componentes básicas de los esquemas GRASP y VNS; del mismo modo, se detallará la manera en que estos algoritmos fueron adaptados para el problema de estudio y el modo en que fueron embebidos para dar paso a un algoritmo híbrido.

## 4.1 GREEDY RANDOMIZED ADAPTIVE SEARCH PROCEDURE

Un esquema GRASP es un algoritmo multiarranque que está compuesto por una fase de construcción de una solución y una de mejora de la misma, este proceso determina una iteración GRASP. Múltiples iteraciones GRASP son ejecutadas hasta alcanzar un criterio de parada. El Algoritmo 1 describe un GRASP que resuelve un problema de minimización con una fase constructiva y una de mejora como las descritas en las Secciones 4.1.1 y 4.1.2.

### 4.1.1 FASE DE CONSTRUCCIÓN

En la primera fase del GRASP se obtienen soluciones del problema haciendo uso de un algoritmo constructivo, es decir, los elementos de la solución se van incorporando uno a la vez de manera iterativa. En problemas de ruteo, los algoritmos constructivos más conocidos son el de ahorros, el de vecino más cercano, el de inserción más cercana, etcétera.

Para decidir qué elemento nuevo será integrado en la solución parcial se define una función miope que mide la contribución local de cada candidato a la calidad de la solución. Usualmente, un algoritmo constructivo elige al candidato con mejor evaluación de su función miope; sin embargo, una construcción tipo GRASP añade cierto grado de aleatoriedad en esta decisión.

La manera más utilizada para añadir aleatoriedad al proceso es el uso de una lista restringida de candidatos LRC, la cual contiene a los candidatos mejor evaluados y de entre ellos se elige aleatoriamente quién será integrado a la solución parcial. Para determinar la cardinalidad de la LRC, generalmente se utiliza un parámetro de calidad denotado por  $\alpha$  tal que  $\alpha \in [0, 1]$ .

Consideremos un problema en el cual entre menor sea la función de evaluación,

mejor es el candidato y sean  $e_i$  el valor de la función de evaluación del candidato  $i$ ;  $e_{min}$ , la mínima función de evaluación y  $e_{max}$ , la máxima. Entonces, la LRC estará conformada por todos los elementos  $i$  tales que  $e_i \in [e_{min}, e_{min} + \alpha(e_{max} - e_{min})]$ .

Nótese que si  $\alpha = 0$ , en la LRC solo estará el candidato mejor evaluado y la decisión de cuál elemento formará parte de la solución es miope o voraz. En caso contrario, si  $\alpha = 1$ , la LRC estará formada por todos los candidatos disponibles y el elemento que se integrará a la solución es elegido en base a una decisión completamente aleatoria.

En consecuencia, determinar un valor de  $\alpha$  apropiado es sumamente importante para obtener soluciones de buena calidad. Una de las propuestas más utilizadas en la literatura para determinar valores adecuados del parámetro de calidad es llamada GRASP reactivo. En un GRASP reactivo, se tiene un conjunto de diferentes valores de  $\alpha$  y para cada iteración GRASP se elige uno de manera aleatoria, siguiendo una distribución de probabilidad que se ajusta automáticamente conforme al desempeño del algoritmo, esto es, si con el uso de algún  $\alpha_0$  se han obtenido buenas soluciones, la probabilidad de elegir  $\alpha_0$  es alta. Este procedimiento se detallará en la Sección 4.1.3.

#### 4.1.2 FASE DE POSTPROCESAMIENTO

Usualmente, para el postprocesamiento se define un vecindario y se realiza una búsqueda local tomando como punto de partida la solución obtenida en la fase de construcción. Sin embargo, recientemente se ha propuesto el uso de metaheurísticas para sustituir a los algoritmos de búsqueda local en la fase de postprocesamiento como lo son búsqueda tabú, recocido simulado, búsqueda por entornos variables y búsqueda local iterada (Festa y Resende, 2009).

**Algoritmo 1** Greedy Randomized Adaptive Search Procedure**Entrada:** $E$  : Conjunto de elementos de la instancia $it_{max}$  : Número de iteraciones GRASP $\alpha$  : Parámetro de calidad $N$  : Vecindario para la búsqueda local

- 1:  $x^* \leftarrow \emptyset$     \\ Solución incumbente
  - 2:  $f^* \leftarrow +\infty$     \\ Valor de la función objetivo de la solución incumbente
  - 3: **para**  $it = 1$  **hasta**  $it = it_{max}$  **hacer**
  - 4:     $x \leftarrow \emptyset$
  - 5:     $C \leftarrow E$
  - 6:    Calcular función miope para todo  $i \in C$
  - 7:    **mientras**  $C \neq \emptyset$  **hacer**
  - 8:      $LRC = \{i \in C | e_i \in [e_{min}, e_{min} + \alpha(e_{max} - e_{min})]\}$
  - 9:     Seleccionar  $i' \in LRC$  de manera aleatoria.
  - 10:     $x \leftarrow x \cup \{i'\}$
  - 11:    Actualizar  $C$
  - 12:    Calcular función miope para todo  $i \in C$
  - 13:    **fin mientras**
  - 14:    **mientras** Existe  $x' \in N(x)$  tal que  $f(x') < f(x)$  **hacer**
  - 15:      $x \leftarrow x'$
  - 16:    **fin mientras**
  - 17:    **si**  $f(x) < f^*$  , **entonces**
  - 18:      $f^* \leftarrow f(x)$
  - 19:      $x^* \leftarrow x$
  - 20:    **fin si**
  - 21: **fin para**
- Salida:**  $x^*, f^*$

## 4.1.3 GRASP REACTIVO

En un esquema GRASP reactivo, se tiene un conjunto  $A = \{\alpha_1, \alpha_2, \dots, \alpha_{|A|}\}$  de posibles valores que puede tomar  $\alpha$ . Al inicio del algoritmo, todos los elementos de  $A$  tienen la misma probabilidad de ser elegidos, esto es  $p_i = \frac{1}{|A|}$ , para  $i = 1, 2, \dots, |A|$ .

Sea  $f^*$  el valor de la función objetivo de la solución incumbente y sea  $\bar{f}_i$  el promedio de las funciones objetivo de las soluciones obtenidas utilizando  $\alpha_i$ . Entonces, de manera periódica se calcula  $p_i = \frac{q_i}{\sum_{j=1}^{|A|} q_j}$ , donde  $q_i = \frac{f^*}{\bar{f}_i}$ , para  $i = 1, 2, \dots, |A|$ .

Nótese que si se obtienen valores de función objetivo bajos utilizando  $\alpha_i$ , entonces  $q_i$  será alto y, en consecuencia, la probabilidad de elegir este valor de  $\alpha$  será mayor.

## 4.2 VARIABLE NEIGHBORHOOD SEARCH

El VNS es una metaheurística propuesta por Mladenović y Hansen (1997) que explota sistemáticamente el cambio de estructuras de vecindarios dentro de un algoritmo de búsqueda local, con el propósito de intensificar y diversificar la búsqueda. Según Hansen y Mladenović (2003), este procedimiento explota los siguientes tres hechos:

1. Un mínimo local con respecto a un vecindario puede no ser el mínimo local con respecto a otro.
2. Un mínimo global es el mínimo local de todos los vecindarios posibles.
3. Muchas veces los óptimos locales con respecto a varios vecindarios están relativamente cerca los unos de los otros, por lo cual un óptimo local provee información acerca del global.

### 4.2.1 VARIABLE NEIGHBORHOOD DESCENT

Un esquema Variable Neighborhood Descent (VND) es un procedimiento metaheurístico perteneciente a la familia de los algoritmos VNS, en el cual se explota la primera de las ideas mencionadas en la sección anterior, la cual menciona que un óptimo local para un vecindario puede no ser el óptimo local en otro.

Sean  $N_1, N_2, \dots, N_l$  estructuras de vecindarios que serán explorados de manera consecutiva en el VND. Sea  $x$  la solución actual y  $x^*$  el óptimo local de  $N_k(x)$  con  $1 \leq k \leq l$ , si dicho óptimo local es mejor que la solución actual, la exploración regresará al primero de los vecindarios que ahora estará centrado en  $x^*$ . El algoritmo termina

cuando no se puede encontrar una mejor solución usando todos los vecindarios, es decir, al final del algoritmo se obtiene el óptimo local para todas las vecindades.

---

**Algoritmo 2** Variable Neighborhood Descent
 

---

**Entrada:** $x$  : Solución inicial $N_1, N_2, \dots, N_l$  : Estructuras de vecindarios1:  $i \leftarrow 1$ 2: **repetir**    **Exploración del vecindario.**3:  $x^* \leftarrow \arg \min_{x' \in N(x)} \{f(x')\}$     **Moverse o no**4:   **si**  $f(x^*) < f(x)$  , **entonces**5:      $x \leftarrow x^*$ 6:      $i \leftarrow 1$ 7:   **si no**8:      $i \leftarrow i + 1$ 9:   **fin si**10: **hasta**  $i = l$ **Salida:**  $x$ 

#### 4.2.2 GENERAL VARIABLE NEIGHBORHOOD SEARCH

Un algoritmo VNS general es un esquema en el cual se explota el uso de múltiples vecindarios para converger a óptimos locales y para guiar la búsqueda hacia regiones promisorias del espacio de soluciones factibles. Dicho algoritmo explota las ideas de que un mínimo local con respecto a un vecindario no necesariamente es el mínimo local con respecto a otro y que los óptimos locales de varios vecindarios se encuentran cerca los unos de los otros.

Para explotar las ideas anteriormente mencionadas, el VNS general hace uso de un esquema VND y de un procedimiento conocido como *agitación (shaking)*, el primero provee intensificación al procedimiento y el segundo, diversificación. Para la agitación de soluciones, se definen estructuras de vecindarios  $N'_1, N'_2, \dots, N'_s$ , que no necesariamente son las mismas que se utilizarán para el VND.

Sean  $x$  la solución actual y  $x'$  una solución aleatoria en  $N'_k(x)$  que será la entrada del esquema VND. Si el óptimo local encontrado es peor que  $x$ , entonces el proceso continúa buscando una solución aleatoria  $x' \in N'_{k+1}(x)$ , si no se regresa al primer vecindario de agitación. Este procedimiento se muestra en el Algoritmo 3.

---

**Algoritmo 3** General Variable Neighborhood Search
 

---

**Entrada:** $x$  : Solución inicial $N_1, N_2, \dots, N_l$  : Estructuras de vecindarios para VND $N'_1, N'_2, \dots, N'_s$  : Estructuras de vecindarios para agitación1:  $j \leftarrow 1$ 2: **repetir**3:   **Agitación:** Encontrar  $x' \in N'_j(x)$  de manera aleatoria.      **Variable Neighborhood Descent:**4:      $i \leftarrow 1$ 5:     **repetir**6:       **Exploración del vecindario:**  $x^* \leftarrow \arg \min_{x'' \in N(x')} \{f(x'')\}$           **Moverse o no:**7:       **si**  $f(x^*) < f(x')$  , **entonces**8:          $x' \leftarrow x^*$ 9:          $i \leftarrow 1$ 10:       **si no**11:          $i \leftarrow i + 1$ 12:       **fin si**13:     **hasta**  $i = l$       **Moverse o no**14:     **si**  $f(x') < f(x)$  , **entonces**15:        $x \leftarrow x'$ 16:        $j \leftarrow 1$ 17:     **si no**18:        $j \leftarrow j + 1$ 19:     **fin si**20: **hasta**  $j = s$ **Salida:**  $x$

### 4.3 ALGORITMO HÍBRIDO TIPO GRASP CON VNS PARA EL OP CON RESTRICCIONES ADICIONALES

El algoritmo heurístico propuesto está basado en un GRASP reactivo, en el cual la fase de mejora de soluciones sigue un esquema VNS general. Cabe mencionar que, usualmente, los algoritmos tipo GRASP carecen de memoria, es decir, las iteraciones son independientes entre sí y no utilizan información obtenida en otras partes del proceso. Sin embargo, existen propuestas en la literatura para incorporar el uso de memoria en un esquema GRASP, las cuales se remontan al trabajo de Fleurent y Glover (1999), en el cual se utiliza una pila de soluciones élite. El algoritmo propuesto en esta tesis, incorpora el uso de memoria mediante una rutina de filtración de soluciones con la cual se busca que las diferentes iteraciones GRASP generen distintas soluciones en la fase de construcción.

En las Secciones 4.3.1 a 4.3.5 explican de manera detallada las componentes del procedimiento propuesto, el cual se muestra en el Algoritmo 4. Se denotará a una instancia del problema como la tupla  $I = (V, M, O, L, t, s, t_{max})$ , cuyas componentes son el conjunto de nodos  $V$ , los conjuntos de nodos obligatorios y opcionales  $M$  y  $O$ , los conjuntos de nodos en conflicto  $L = \{C'_1, C'_2, \dots, C'_k\}$ , la matriz de tiempos de viaje internodales  $t$ , el vector  $s$  de las puntuaciones asociadas a cada nodo y el tiempo máximo  $t_{max}$ . Por otro lado consideraremos a  $r$  como una secuencia de nodos que determinan una ruta, por lo cual  $i \in r$  denota un nodo perteneciente a la ruta  $r$  y, por simplicidad de notación, se utilizará la expresión  $(i, j) \in r$  para referirse a los nodos  $i, j \in r$  tales que  $i$  precede a  $j$  en la ruta.

**Algoritmo 4** Híbrido GRASP-VNS para el OP con restricciones adicionales**Entrada:**

$I = (V, M, O, L, t, s, t_{max})$  : Instancia del problema  
 $A$  : Conjunto de posibles valores de  $\alpha$   
 $it_{max}$  : Número máximo de iteraciones  
 $it_{detener}$  : Número máximo de iteraciones consecutivas sin actualizar el incumbente  
 $it_{actualizar}$  : Periodo de actualización de la distribución de probabilidad de  $A$

- 1:  $r^* \leftarrow \emptyset$  Mejor solución encontrada hasta el momento
- 2:  $s^* \leftarrow 0$  Valor de la función objetivo de  $r^*$
- 3:  $j = 0$
- 4: **para**  $i = 1$  **a**  $it_{max}$  **hacer**
- 5:    $r_i \leftarrow \emptyset$
- 6:    $r_i \leftarrow \text{Construcción}(r, A, G, t)$
- 7:   **si**  $(s_i, d_i) \neq (s_{i'}, d_{i'}), \forall i' < i$  , **entonces**   //  $s_i$  y  $d_i$  son la puntuación y la duración de la ruta de la iteración  $i$
- 8:      $r_i \leftarrow \text{VNS}(r, G, t)$
- 9:     **si**  $s_i > s^*$  , **entonces**
- 10:        $r^* \leftarrow r_i$
- 11:        $j \leftarrow 0$
- 12:     **si no**
- 13:        $j \leftarrow j + 1$
- 14:     **fin si**
- 15: **si no**
- 16:    $j \leftarrow j + 1$
- 17: **fin si**
- 18: **si**  $i \bmod it_{actualizar} = 0$  , **entonces**
- 19:    $A \leftarrow \text{Actualizar}_A(A, s^*)$
- 20: **fin si**
- 21: **si**  $j = it_{detener}$  , **entonces**
- 22:   **break**
- 23: **fin si**
- 24: **fin para**

**Salida:**  $r^*$

## 4.3.1 CONSTRUCCIÓN DE SOLUCIONES

El algoritmo de construcción de soluciones está dividido en dos etapas, la primera comprende la generación de una ruta parcial que solamente contiene a los nodos obligatorios, y en la segunda se agregan algunos nodos opcionales a la solución generada en la primera fase; en ambas etapas se utiliza el mismo valor del parámetro

$\alpha$ .

#### FASE DE CONSTRUCCIÓN CON NODOS OBLIGATORIOS

La etapa del algoritmo de construcción correspondiente a los nodos obligatorios sigue un esquema basado en un algoritmo de inserción más cercana como se muestra en el Algoritmo 5.

Primero, se crea una ruta parcial que solamente contiene a los depósitos (nodos 1 y  $n$ ), la cual está denotada por  $r$ . Si en la instancia solo se tiene un depósito, se creará un nodo artificial  $n + 1$  cuyas coordenadas son las mismas que las del nodo 1, y la ruta parcial contendrá a estos dos.

Si  $t_{ij}$  es el tiempo requerido para viajar del nodo  $i$  al nodo  $j$ , la función de evaluación para cada nodo  $k$  obligatorio es equivalente al mínimo aumento en la longitud de la ruta generado al insertar  $k$  en la solución parcial y está dada por la expresión (4.1).

$$e_k = \min_{(i,j) \in r} \{t_{ik} + t_{kj} - t_{ij}\} \quad (4.1)$$

En seguida, se elige de manera aleatoria el valor del parámetro  $\alpha \in A$  para generar una lista restringida de candidatos (LRC) definida por la expresión (4.2), donde  $e_{min}$  y  $e_{max}$  son el mínimo y el máximo de los valores de las funciones de evaluación de los candidatos, respectivamente. Después, se elige un nodo de la LRC de manera aleatoria y este es insertado en la ruta en la posición en la cual el aumento en la longitud del recorrido es mínimo.

$$LRC = \{k \in V | e_k \leq (1 - \alpha)e_{min} + \alpha e_{max}\} \quad (4.2)$$

A continuación, se vuelven a calcular los valores de la función de evaluación para todos los nodos obligatorios que no se han incorporado a la ruta y el proceso

se repite hasta que la solución contiene a todos los nodos obligatorios o hasta que no se pueda agregar un nodo sin que se viole la restricción de la duración máxima de la ruta.

En el caso de que existan nodos obligatorios que no se puedan incluir en la solución, estos se incorporarán al conjunto de nodos opcionales, asignándoles una puntuación igual a  $\sum_{i \in O} s_i$ .

---

**Algoritmo 5** Fase de construcción con nodos obligatorios
 

---

**Entrada:**

$I = (V, M, O, L, t, s, t_{max})$  : Instancia del problema

$A$  : Conjunto de posibles valores de  $\alpha$

- 1:  $r \leftarrow \{1, n\}$
- 2:  $LC \leftarrow M$
- 3:  $M' \leftarrow \emptyset$
- 4: **mientras**  $LC \neq \emptyset$  **hacer**
- 5:    $e_k \leftarrow \min\{t_{ik} + t_{kj} - t_{ij} \mid i, j \in r \text{ tal que } i \text{ es adyacente a } j\}, \forall k \in LC$
- 6:    $e_{min} \leftarrow \min\{e_k \mid k \in LC\}$
- 7:    $e_{max} \leftarrow \max\{e_k \mid k \in LC\}$
- 8:    $LRC \leftarrow \{k \in LC \mid e_k \leq (1 - \alpha)e_{min} + \alpha e_{max}\}$
- 9:   Seleccionar  $k' \in LRC$  aleatoriamente
- 10:   **si**  $d(r) + e_{k'} \leq t_{max}$ , **entonces**
- 11:     Insertar  $k'$  en  $r$  en la posición correspondiente a la inserción más cercana
- 12:   **si no**
- 13:      $M' \leftarrow M' \cup k'$
- 14:   **fin si**
- 15:    $LC \leftarrow LC \setminus \{k'\}$
- 16: **fin mientras**
- 17: **si**  $M' \neq \emptyset$ , **entonces**
- 18:    $s_k \leftarrow \sum_{i \in O} s_i, \forall k \in M'$
- 19:    $O \leftarrow O \cup M'$
- 20:    $M \leftarrow M \setminus M'$
- 21: **fin si**

**Salida:**  $r$

---

## FASE DE CONSTRUCCIÓN CON NODOS OPCIONALES

La fase de construcción relativa a los nodos opcionales sigue un esquema similar al de la relacionada con los obligatorios, la diferencia entre ambas etapas reside en la

función de evaluación de los nodos y en la actualización del conjunto de candidatos, como se muestra en el Algoritmo 6.

Inicialmente, la lista de candidatos (LC) estará conformada por todos los nodos opcionales, para los cuales se calcula la función de evaluación  $e'_k$  definida por la expresión (4.3), donde  $c$  es el número de nodos que tienen conflicto con el nodo  $k$  y  $e_k$  está definido como en la ecuación (4.1).

$$e'_k = \frac{s_k + 1}{(c + 1)(e_k)} \quad (4.3)$$

En este caso, se consideran el costo mínimo de inserción, el beneficio asociado al candidato y el número de nodos con los que este tiene conflicto.

Posteriormente se construye una lista restringida de candidatos (LRC') utilizando el mismo valor de  $\alpha$  que fue usado en la etapa referente a los nodos obligatorios y que está definida por la expresión (4.4). De la LRC' se selecciona un nodo de manera aleatoria para ser incorporado a la ruta parcial en la posición en la cual el aumento en la duración de la ruta es mínimo.

$$LRC' = \{k \in LC | e'_k \geq (1 - \alpha)e'_{max} + \alpha e'_{min}\} \quad (4.4)$$

Finalmente, la nueva lista de candidatos estará conformada por los nodos opcionales que no han sido incluidos en la ruta y que no tienen conflicto con ningún elemento ya incluido en la solución parcial. Se recalculan los valores de la función de evaluación para cada nodo de la LC y el proceso se repite de manera iterativa hasta que no se puede insertar un nuevo nodo sin que la restricción de duración máxima de la ruta sea violada.

**Algoritmo 6** Fase de construcción con nodos opcionales**Entrada:** $I = (V, M, O, L, t, s, t_{max})$  : Instancia del problema $A$  : Conjunto de posibles valores de  $\alpha$  $r$  : Solución construida en la fase de nodos obligatorios $LC \leftarrow O$ 1: **mientras**  $LC \neq \emptyset$  **hacer**2:  $e_k \leftarrow \min\{t_{ik} + t_{kj} - t_{ij} | i, j \in r \text{ tal que } i \text{ es adyacente a } j\}, \forall k \in LC$ 3:  $e'_k \leftarrow \frac{s_k + 1}{(c+1)(e_k)}, \forall k \in LC$ 4:  $e'_{min} \leftarrow \min\{e'_k | k \in LC\}$ 5:  $e'_{max} \leftarrow \max\{e'_k | k \in LC\}$ 6:  $LRC \leftarrow \{k \in LC | e_k \geq (1 - \alpha)e'_{max} + \alpha e'_{min}\}$ 7: Seleccionar  $k' \in LRC$  aleatoriamente8: **si**  $d(r) + e_{k'} \leq t_{max}$ , **entonces**9:     Insertar  $k'$  en  $r$  en la posición correspondiente a la inserción más cercana10:      $LC \leftarrow LC \setminus (\{k'\} \cup C_{k'})$ 11: **fin si**12: **fin mientras****Salida:**  $r$ 

## 4.3.2 FILTRADO DE SOLUCIONES

El propósito de una construcción tipo GRASP, es el de obtener soluciones diversas de buena calidad; no obstante, puede llegar a ocurrir que en iteraciones distintas se generen soluciones idénticas. Para evitar esta situación, en el algoritmo propuesto en esta tesis se incorpora una rutina de filtración de soluciones.

Para prevenir que se almacenen todas las soluciones construidas en las iteraciones GRASP previas, el algoritmo de filtro solo guarda pares ordenados  $(s, d)$  en donde la primera componente está asociada a la puntuación y a la segunda a la duración de una ruta.

Sean  $R$  el conjunto de pares ordenados almacenados y  $(s_r, d_r)$  la dupla asociada a una nueva solución  $r$ . Si  $(s_r, d_r)$  es diferente a cada par contenido en  $R$ , entonces este par es guardado y pasa a la fase de mejora; de lo contrario, la solución se desecha y se termina la iteración GRASP.

### 4.3.3 VECINDARIOS UTILIZADOS EN EL VNS

Para la etapa de agitación de soluciones y para la búsqueda local del VNS general, se proponen tres tipos de vecindario definidos por:

- **N1.** Intercambio entre un nodo que pertenece a la solución y otro que no ha sido incorporado en ella.
- **N2.** Inserción de un nuevo nodo en la ruta.
- **N3.** Reordenamiento de la solución.

#### VECINDARIO DEFINIDO POR INTERCAMBIOS

Un vecino de la solución actual en N1 será cualquier ruta resultante al realizar los siguientes movimientos:

1. Seleccionar de manera aleatoria un nodo  $n \in r$  y removerlo.
2. Si  $n$  es opcional, elegir un nodo  $n'$  que no forme parte de la solución y que no tenga conflicto con ningún nodo en  $r$ . Si no,  $n' \leftarrow n$ .
3. Insertar a  $n'$  en la posición en la cual el incremento en la duración de la ruta es mínimo.

El caso en que  $n$  es opcional es ilustrado en la Figura 4.1b y la Figura 4.1c muestra el caso en el que  $n$  es obligatorio y este solo es reinsertado en la solución.

#### VECINDARIO DEFINIDO POR INSERCIONES

El movimiento que define a esta vecindad consiste en elegir un nodo que no forme parte de la ruta actual e incorporarlo en la posición correspondiente al mínimo aumento en la duración de la misma, como se muestra en la Figura 4.1d.

Para elegir al vecino aleatorio en N2 en la fase de agitación, se calcula una probabilidad de elección  $p(k) = \frac{e'_k}{\sum_{i \in V \setminus \cup \{j \in r\}} e'_i}$  para todo nodo  $k$  que no forma parte de la solución, donde  $e'_k$  está definido como en la expresión (4.3).

#### VECINDARIO DEFINIDO POR REORDENAMIENTO

Una solución vecina de la actual en N3, estará definida por el resultado de llevar a cabo los movimientos siguientes:

1. Elegir una cadena definida por un conjunto de nodos consecutivos en la solución actual, sin incluir a los depósitos inicial y final, y removerla, como se muestra en la Figura 4.1e.
2. Escoger un par de nodos consecutivos de la nueva ruta y reinsertar la cadena entre ellos, tal como se ilustra en la Figura 4.1f.

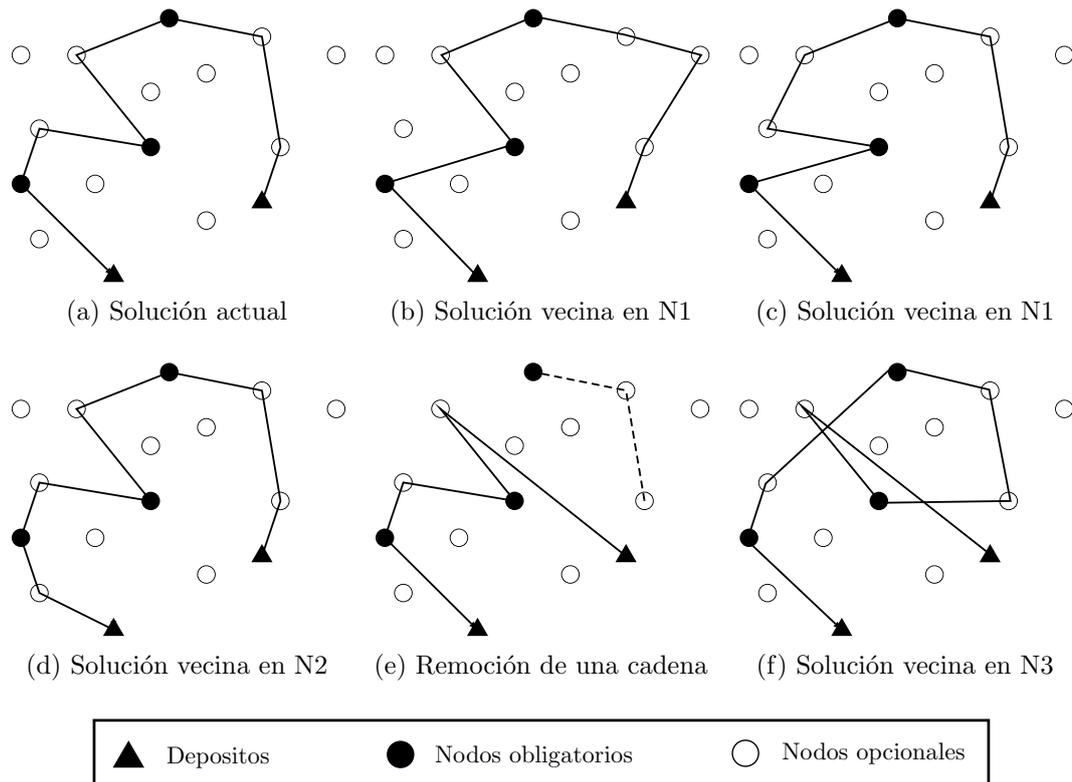


Figura 4.1: Vecindarios utilizados en el VNS

#### 4.3.4 RECUPERACIÓN DE FACTIBILIDAD

Cabe hacer la observación de que al moverse a un vecino en N1, N2 o N3, dicha solución puede ser infactible, ya sea porque se exceda la duración máxima permitida o porque existan incompatibilidades entre nodos de la ruta. En tal caso, la factibilidad es recuperada de la siguiente manera:

- **Infactibilidad debido a violación de la duración máxima de la ruta.** El nodo opcional con la menor puntuación es removido de la solución, este proceso se repite hasta que la solución es factible, tal como se muestra en la Figura 4.2.
- **Infactibilidad debido a la existencia de conflictos.** El nodo que es incom-

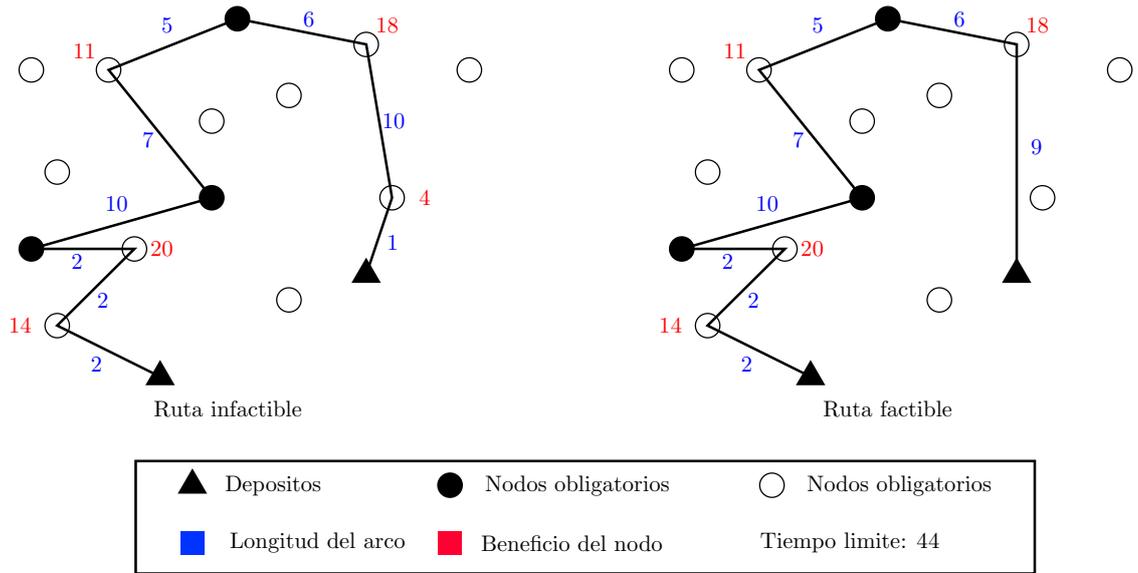


Figura 4.2: Recuperación de factibilidad debido a la violación de la duración máxima

patible con el que recién se integró a la solución es removido, como se muestra en la Figura 4.3.

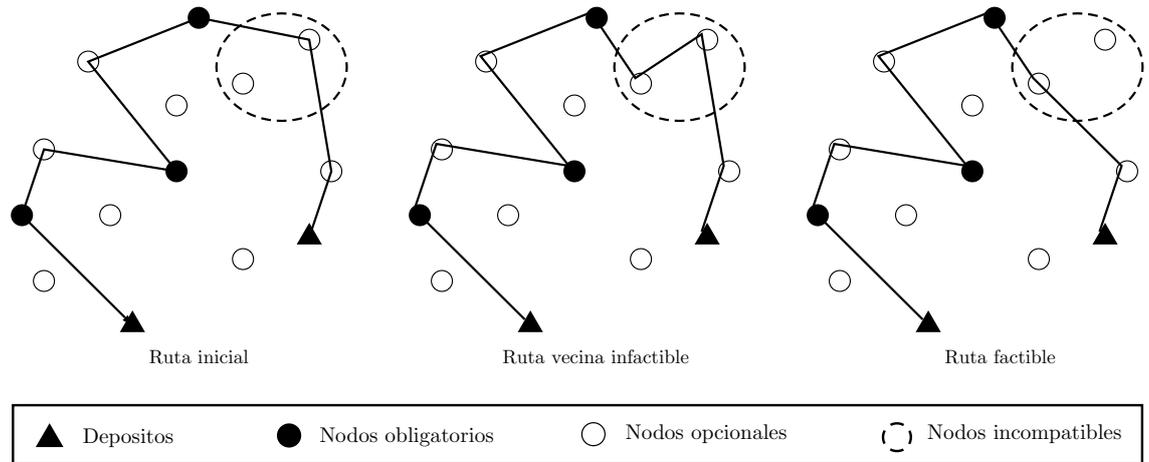


Figura 4.3: Recuperación de factibilidad debido a existencia de conflictos

### 4.3.5 BÚSQUEDA LOCAL POR VND

De acuerdo con el Algoritmo 2, los vecindarios deben ser explorados por completo en la fase de búsqueda local del VND; sin embargo, cuando la vecindad es muy grande, esta estrategia puede ser ineficiente, debido al alto costo computacional

requerido.

Dado lo anterior, en el procedimiento tipo VND que se propone en esta tesis, en lugar de examinar los vecindarios completos, se definen estrategias de exploración mediante tres algoritmos diferentes de búsqueda local que utilizan las vecindades N1 y N2, los cuales se describen en las secciones siguientes.

#### ALGORITMO DE BÚSQUEDA LOCAL 1

El primero de los algoritmos de búsqueda local, denotado por LS1, explora el espacio de soluciones mediante el uso del vecindario de inserciones N2 y selecciona los nodos a insertar de una lista restringida de candidatos.

En primera instancia, para cada nodo que no ha sido incorporado en la solución actual, se calcula la función (4.3) y se crea una lista restringida de candidatos definida por (4.4), utilizando el mismo valor de  $\alpha$  que se usó para la fase de construcción. Un nodo aleatorio es seleccionado de la LRC y se inserta en la ruta en la posición en la cual el incremento en la duración del recorrido es mínimo. Finalmente, se recupera la factibilidad de la solución, si es necesario.

En este algoritmo la solución es actualizada al encontrar al primer vecino de la solución actual, aunque no sea implique una mejora en el incumbente. El procedimiento termina cuando se alcanza un número máximo de iteraciones consecutivas sin mejora y ,finalmante, se reporta la mejor solución encontrada a lo largo del método. En esta implementación, el número máximo de iteraciones consecutivas sin mejora se fijó en 10 como resultado de experimentación previa.

El Algoritmo 7 muestra la manera en que opera el procedimiento LS1.

**Algoritmo 7** Búsqueda local 1**Entrada:** $r$  : Solución actual $I = (V, M, O, L, t, s, t_{max})$  : Instancia del problema

- 1:  $r' \leftarrow r$
- 2:  $r^* \leftarrow r$
- 3:  $LC \leftarrow O \setminus \cup \{i | i \in r'\}$
- 4:  $it \leftarrow 0$
- 5: **mientras**  $LC \neq \emptyset \wedge it < 10$  **hacer**
- 6:  $e'_k \leftarrow \frac{s_k+1}{(c+1)(e_k)}, \forall k \in LC$
- 7:  $e'_{min} \leftarrow \min\{e'_k | k \in LC\}$
- 8:  $e'_{max} \leftarrow \max\{e'_k | k \in LC\}$
- 9:  $LRC \leftarrow \{k \in O | e'_k \leq (1 - \alpha)e'_{min} + \alpha e'_{max}\}$
- 10: Seleccionar  $k' \in LRC$  aleatoriamente
- 11: Insertar  $k'$  en  $r'$  en la posición acorde a la inserción más cercana
- 12: Recuperar factibilidad de  $r'$
- 13:  $LC \leftarrow O \setminus \cup \{i | i \in r'\}$
- 14: **si**  $s(r') > s(r^*)$ , **entonces**
- 15:  $r^* \leftarrow r'$
- 16:  $it \leftarrow 0$
- 17: **si no**
- 18:  $it \leftarrow it + 1$
- 19: **fin si**
- 20: **fin mientras**

**Salida:**  $r^*$ 

## ALGORITMO DE BÚSQUEDA LOCAL 2

El segundo algoritmo de búsqueda local (LS2) explora el espacio de soluciones a través del vecindario N2 definido por inserciones de nodos que no pertenecen a la solución actual. El procedimiento es ilustrado por el Algoritmo 8.

En primera instancia, los nodos que no han sido incorporados en la solución son evaluados de acuerdo a la función (4.3) y, de acuerdo a este valor, son ordenados de manera decreciente. Este ordenamiento dictará el orden en que será explorado el vecindario.

Se elige el primero de los nodos, i.e. el que posee el mayor valor de la función de

evaluación, para posteriormente insertarlo en la posición correspondiente al menor aumento en la duración de la ruta y recuperar la factibilidad de la solución en el caso de que se requiera.

Si el algoritmo encuentra una mejor solución que la actual, esta última será sustituida por la nueva, se recalculan las funciones de evaluación y el proceso es repetido; si no, se intentará insertar el nodo siguiente con respecto al orden dado por la función de evaluación. El proceso termina cuando se ha explorado todo el vecindario de la solución actual y no se han encontrado mejoras.

---

**Algoritmo 8** Búsqueda local 2
 

---

**Entrada:**

- $r$  : Solución actual
  - $I = (V, M, O, L, t, s, t_{max})$  : Instancia del problema
  - 1:  $LC \leftarrow O \setminus \cup \{i | i \in r\}$
  - 2: **mientras**  $LC \neq \emptyset$  **hacer**
  - 3:    $r' \leftarrow r$
  - 4:    $e'_k \leftarrow \frac{s_k+1}{(c+1)(e_k)}, \forall k \in LC$
  - 5:    $k' \leftarrow \arg \max_{k \in LC} \{e'_k\}$
  - 6:   Insertar  $k'$  en  $r'$  en la posición acorde a la inserción más cercana
  - 7:   Recuperar factibilidad de  $r'$
  - 8:   **si**  $s(r') > s(r)$  , **entonces**
  - 9:      $r \leftarrow r'$
  - 10:    $LC \leftarrow O \setminus \cup \{i | i \in r\}$
  - 11:   **si no**
  - 12:      $LC \leftarrow LC \setminus \{k'\}$
  - 13:   **fin si**
  - 14: **fin mientras**
- 

## ALGORITMO DE BÚSQUEDA LOCAL 3

El último de los algoritmos de búsqueda local utilizado por el VND, denotado por LS3, usa como base al vecindario N1, definido por intercambios de nodos y se muestra en el Algoritmo 9. La manera en que se eligen a las soluciones vecinas a ser evaluadas y el orden en que se inspeccionan se explican a continuación.

Sea  $r[k]$  el nodo que se encuentra en la  $k$ -ésima posición en la ruta  $r$ , donde  $k = 2, \dots, |r| - 1$ , ( $k = 1$  y  $k = |r|$  corresponden a los depósitos inicial y final, respectivamente). En una iteración de este algoritmo, se removerá  $r[k]$  y se insertará un nodo aleatorio de acuerdo a la estructura de N1.

Si el nodo que se ha seleccionado para abandonar la solución es opcional y la lista de candidatos es vacía, el elemento removido será reinsertado en la ruta de acuerdo a su inserción más cercana. En caso de que se requiera, se recupera la factibilidad de la solución.

Si la solución obtenida después realizar los movimientos anteriores es mejor que el incumbente, este es actualizado y se fija  $k = 2$ ; si no,  $k = k + 1$ . El algoritmo se detiene cuando  $k = |r|$ .

---

**Algoritmo 9** Búsqueda Local 3

---

**Entrada:**

$r$  : Solución actual  
 $I = (V, M, O, L, t, s, t_{max})$  : Instancia del problema  
1:  $i \leftarrow 2$   
2:  $r' \leftarrow r$   
3: **mientras**  $i < |r'|$  **hacer**  
4:    $k \leftarrow r'[i]$   
5:    $r' \leftarrow r' \setminus \{k\}$   
6:   **si**  $k \in M$  , **entonces**  
7:      $k' \leftarrow k$   
8:   **si no**  
9:      $LC \leftarrow (O \setminus \bigcup_{i \in r} C_i) \cup C_k$   
10:    **si**  $LC \neq \emptyset$  , **entonces**  
11:     Seleccionar  $k' \in LC$  aleatoriamente  
12:    **si no**  
13:      $k' \leftarrow k$   
14:    **fin si**  
15:   **fin si**  
16:   Insertar  $k'$  en  $r'$  en la posición acorde a la inserción más cercana  
17:   Recuperar factibilidad de  $r'$   
18:   **si**  $s(r') > s(r)$  , **entonces**  
19:      $r \leftarrow r'$   
20:      $i \leftarrow 2$   
21:   **si no**  
22:      $i \leftarrow i + 1$   
23:   **fin si**  
24: **fin mientras**

---

En el siguiente capítulo se presenta la experimentación computacional sobre un gran conjunto de instancias. La calidad de los resultados obtenidos muestran la efectividad del algoritmo propuesto.

## CAPÍTULO 5

# RESULTADOS COMPUTACIONALES DEL ALGORITMO HEURÍSTICO

---

Cuando se implementa un algoritmo heurístico es importante la elección adecuada de los parámetros que se utilizarán, ya que de ello depende el desempeño del algoritmo. En particular, en un GRASP reactivo se deben tomar decisiones acerca del periodo de actualización de la probabilidad de elección para cada valor  $\alpha$  y el criterio de paro (que puede ser número de iteraciones, tiempo de ejecución, número de iteraciones consecutivas sin mejora, etcétera). Por otro lado, en un VNS general se deberá decidir el orden en que se explorarán los vecindarios de agitación y la secuencia en que serán ejecutados los algoritmos de búsqueda local en el VND.

Para tomar estas decisiones, se realizó experimentación computacional previa sobre un gran conjunto de instancias, para posteriormente llevar a cabo pruebas de hipótesis que den soporte estadístico a la elección de los parámetros. Dichas pruebas, que fueron ejecutadas en el software estadístico Minitab 17, se presentan en este capítulo. Además, se presentan los resultados computacionales obtenidos de la ejecución del algoritmo híbrido GRASP-VNS.

## 5.1 ORDEN DE LA EJECUCIÓN DE LOS ALGORITMOS DE BÚSQUEDA LOCAL EN EL VND

De acuerdo con Hansen y Mladenović (2003), una práctica común para determinar el orden en que los algoritmos de búsqueda local serán ejecutados en el esquema VND, consiste en ejecutar dichos algoritmos en orden ascendente con respecto al tamaño de los vecindarios explorados por ellos.

En el procedimiento metaheurístico propuesto en esta tesis, se siguió dicha metodología, para lo cual fue necesario, en primera instancia, determinar el tamaño de los vecindarios N1 y N2.

### 5.1.1 TAMAÑO DEL VECINDARIO N1

El vecindario N1 está definido por intercambios entre nodos que se encuentran en la ruta y nodos que no pertenecen a ella y que, además, no tienen conflictos con los elementos ya incluidos.

Considere una solución factible  $r$  y sea  $q$ , el número de nodos que no están incluidos en ella, de manera que  $|r| + q = |V|$ . El número de nodos que pueden salir de la ruta es igual a  $|r| - 2$  y esta cantidad está acotada por  $|r|$ . Por otro lado, para cada elemento  $k$  que puede salir de la solución, el número de nodos que pueden entrar a ella está dado por la expresión (5.1).

$$q - \sum_{k \in r} c_k \tag{5.1}$$

Donde  $c_k$  es el número de nodos con los cuales  $k$  tiene conflicto.

Ahora bien, por las características de las instancias, cada nodo puede estar libre de conflictos o ser incompatible con un máximo de tres nodos. Por lo tanto, se cumple la expresión (5.2).

$$0 \leq \sum_{k \in r} c_k \leq 3|r| \quad (5.2)$$

De las expresiones (5.1) y (5.2) se obtiene la siguiente desigualdad.

$$q - 3|r| \leq q - \sum_{k \in r} c_k \leq q \quad (5.3)$$

Finalmente, una cota superior para el número de vecinos de  $r$  en  $N1$ , está dado por la siguiente expresión.

$$|N1(r)| \leq |r|q \quad (5.4)$$

Por lo tanto  $|N1(r)| = O(|r|q)$ .

### 5.1.2 TAMAÑO DEL VECINDARIO N2

La vecindad N2 está definida por inserciones de nodos que no han sido incorporados en la solución, sin importar si estos son incompatibles con un nodo que forma parte de la ruta o no. Por lo tanto, es evidente que  $|N2(r)| = O(q)$ , donde  $q$  está definido como en la sección anterior.

### 5.1.3 ORDEN DE LOS ALGORITMOS DE BÚSQUEDA LOCAL

Tal y como se describió en la Sección 4.3.5, se tienen tres algoritmos de búsqueda local, denotados por L1, L2 y L3. L1 y L2 exploran el espacio de soluciones haciendo uso del vecindario N2 y L3, a través de N1.

Debido a que L3 hace uso del vecindario de mayor cardinalidad, es claro que este será el último de los algoritmos a ejecutarse. Sin embargo, se necesita un análisis adicional para determinar el orden en que L1 y L2 serán llamados por el VND.

Cabe destacar que, el conjunto de los candidatos a entrar en la solución en el algoritmo L1 es un subconjunto del vecindario N1, ya que en este procedimiento se utiliza una lista restringida de candidatos para la selección del nodo que entrará a la solución. Por otro lado, los candidatos a formar parte de la solución en L2 son todos los nodos contenidos en N1.

Como consecuencia de lo anterior, el orden en que se aplican los algoritmos de búsqueda local en el VND es: primero L1, seguido de L2 y, finalmente, L3.

## 5.2 ORDEN DE LA EXPLORACIÓN DE LOS VECINDARIOS DE AGITACIÓN EN EL VNS

Con el propósito de determinar el orden apropiado en que los vecindarios de agitación del VNS deben ser explorados, se ejecutaron seis versiones distintas del algoritmo híbrido GRASP-VNS para cada una de las instancias de prueba. Las diferentes versiones del algoritmo corresponden a cada posible permutación en el orden de exploración de las vecindades de agitación como se indica en la Tabla 5.1.

<b>Versión</b>	<b>Orden de los vecindarios de agitación</b>
1	N1, N2, N3
2	N1, N3, N2
3	N2, N1, N3
4	N2, N3, N1
5	N3, N1, N2
6	N3, N2, N1

Tabla 5.1: Versiones del algoritmo GRASP-VNS

Con el propósito de realizar comparaciones consistentes entre la calidad de las soluciones obtenidas por las distintas versiones del algoritmo, para cada solución reportada se calculó el gap con respecto al mejor valor de la función objetivo encontrado en esa instancia.

Para seleccionar la versión del algoritmo que provee soluciones de mayor calidad, se consideró a cada una de ellas como un tratamiento distinto y al gap de cada resultado obtenido como una observación y se realizó una prueba no paramétrica de Kruskal Wallis cuya hipótesis nula  $H_0$  consiste en que no existen diferencias significativas entre las medianas de las observaciones para cada tratamiento y cuya hipótesis alternativa  $H_1$  se basa en que, al menos una de las medianas, es distinta a las demás, lo cual se representa en las ecuaciones (5.5) y (5.6).

$$H_0 : Me_1 = Me_2 = Me_3 = Me_4 = Me_5 = Me_6 \quad (5.5)$$

$$H_1 : Me_i \neq Me_j, \text{ para } i \neq j, i \quad (5.6)$$

En donde  $Me_i$  denota la mediana de las observaciones obtenidas al ejecutar la versión  $i$  del algoritmo GRASP-VNS.

Al realizar la prueba de Kruskal Wallis se obtuvieron los resultados que se muestran en la Tabla 5.2, de la cual se concluye que no se puede rechazar la hipótesis nula. Por lo tanto, con un valor  $p$  de 0.682 se puede asegurar que las medianas de los gaps obtenidos por cada versión del algoritmo no presentan diferencias significativas y, en consecuencia, el orden en que se apliquen los vecindarios de agitación en el VNS no afecta en la calidad de las soluciones obtenidas.

<b>Version GRASP-VNS</b>	<b>N</b>	<b>Mediana</b>	<b>Clasificación del promedio</b>	<b>Z</b>
1	340	0.000000000	996.4	-0.83
2	340	0.000000000	1012.9	-0.26
3	340	0.000000000	990.7	-1.02
4	340	0.000000000	1033.6	0.45
5	340	0.000000000	1032.2	0.40
6	340	0.000000000	1057.1	1.26
General	2040		1020.5	
H=3.12	GL=5	P=0.682		

Tabla 5.2: Prueba de Kruskal Wallis: Gap de la función objetivo contra versión GRASP-VNS

Debido a lo anterior, se prosiguió a buscar la versión de la metaheurística que obtiene las soluciones en un menor tiempo de ejecución. Se realizó una prueba similar para los gaps obtenidos para el tiempo de ejecución con la hipótesis nula  $H_0$  de que no existen diferencias estadísticamente significativas entre las medianas de los gaps de los tiempos de ejecución observados entre cada tratamiento contra la hipótesis alternativa  $H_1$  de que, al menos, una de las medianas es distinta. Dichas hipótesis se representan por las ecuaciones (5.5) y (5.6).

<b>Version GRASP-VNS</b>	<b>N</b>	<b>Mediana</b>	<b>Clasificación del promedio</b>	<b>Z</b>
1	340	0.01844	685.4	-11.49
2	340	0.05295	831.1	-6.50
3	340	0.03752	738.7	-9.66
4	340	0.08035	991.9	-0.98
5	340	0.20185	1434.9	14.21
6	340	0.20689	1441.1	14.42
General	2040		1020.5	
H = 568.12	GL = 5	P = 0.000		

Tabla 5.3: Prueba de Kruskal Wallis: Gap del tiempo contra versión GRASP-VNS

De la Tabla 5.3 se puede concluir que se rechaza la hipótesis nula en favor de la alternativa de que al menos una de las medianas de los tratamientos es sig-

nificantivamente diferente a las demás. Es decir, al menos una de los versiones de los algoritmos requiere un menor tiempo de cómputo para converger a la solución óptima local.

Con el propósito de determinar cuál de las versiones es la que converge en un menor tiempo, se calcularon los intervalos de confianza para las medias utilizando el ajuste de Bonferroni, con una confiabilidad del 95 %, los cuales se muestran en la Tabla 5.4 y se pueden apreciar de manera gráfica en la Figura 5.1.

Version GRASP-VNS	Límite inferior	Límite superior
1	0.050962	0.0852039
2	0.05941	0.082301
3	0.0498834	0.0740526
4	0.0817702	0.106161
5	0.173731	0.205008
6	0.175096	0.206368

Tabla 5.4: Intervalos de confianza para la media del gap de cada versión del GRASP-VNS con una confiabilidad del 95 %

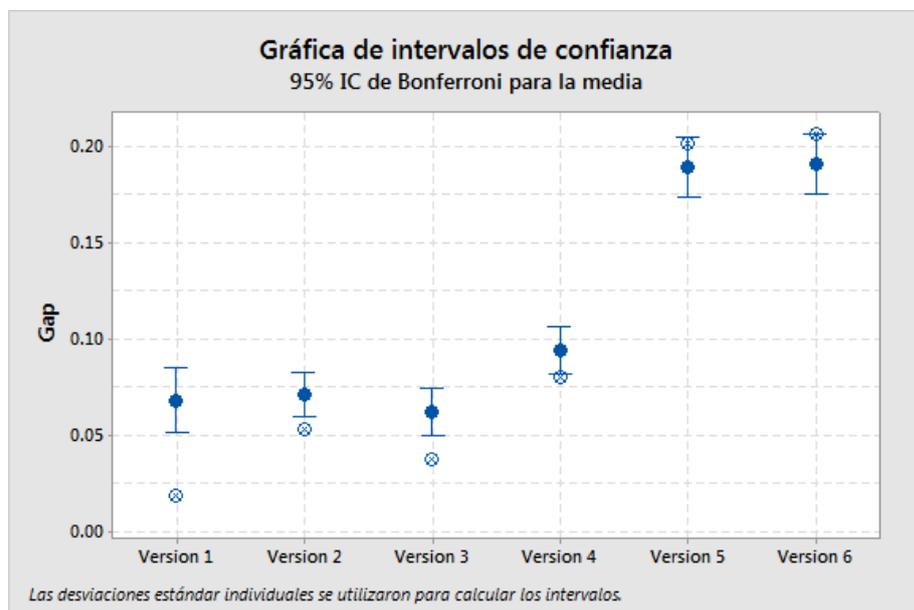


Figura 5.1: Intervalos de confianza para la media del gap de cada versión del GRASP-VNS con una confiabilidad del 95 %

Es evidente que las versiones 5 y 6 del algoritmo híbrido muestran un gap para

el tiempo de ejecución significativamente mayor que el de las versiones 1, 2, 3 y 4. Además, también hay diferencias significativas entre la 3 y la 4.

Sin embargo, no se puede concluir que los tiempos requeridos por las versiones 1, 2 y 3 sean distintos. Por lo tanto, cualquiera de ellas podría ser tomada como la que requiere menor tiempo de cómputo. Finalmente, se eligió la versión 1 del algoritmo, en el cual el orden de exploración de los vecindarios de agitación es N1, N2 y N3.

### 5.3 PARÁMETROS UTILIZADOS EN EL GRASP

Como se mencionó anteriormente, el uso de un esquema GRASP reactivo evita el trabajo de calibrar el parámetro de calidad  $\alpha$  utilizado en la fase de construcción del mismo. Aún así, debe establecerse un criterio de paro del algoritmo y el periodo de actualización de la probabilidad de elección para cada posible valor de  $\alpha$ .

En el algoritmo que se propone en este trabajo, la actualización de la distribución de probabilidad del conjunto  $A$  se realiza cada 100 iteraciones GRASP, debido a que este valor es utilizado ampliamente en la literatura.

Por otro lado, por experimentación previa en las instancias de mayor cardinalidad, se determinó un criterio de paro del algoritmo de 1000 iteraciones; sin embargo, el número requerido por el método para converger a un óptimo local en instancias más pequeñas es menor. Por lo tanto, con el objetivo de fijar un criterio de parada del algoritmo que sea más apropiado para cada instancia, se optó por detener el procedimiento una vez que se alcance un cierto número de iteraciones consecutivas sin actualizar la solución incumbente.

Para fijar de manera adecuada el número de iteraciones consecutivas sin actualizar la solución incumbente que deben alcanzarse para detener el algoritmo se llevó a cabo experimentación computacional sobre las 340 instancias de prueba. En primera instancia se ejecutó el algoritmo con 1000 iteraciones GRASP. Posterior-

mente, se realizaron experimentos deteniendo la ejecución del método al alcanzar 100, 200, 300, 400 y 500 iteraciones consecutivas sin mejorar el incumbente.

En seguida, para cada una de estas versiones del GRASP se calculó la cantidad de instancias en las cuales se obtuvo el mejor valor de la función objetivo. En la Tabla 5.5 se muestra el número de veces que cada versión del algoritmo alcanzó la mejor solución. Nótese que las versiones de la metaheurística que alcanzan la mejor solución en más ocasiones son aquellas que se detienen al alcanzar 500 iteraciones sin actualizar la solución incumbente y 1000 iteraciones GRASP.

<b>Criterio de parada</b>	<b>Veces que se alcanzó la mejor solución</b>
100	213
200	222
300	239
400	242
500	244
1000*	244

*\*El criterio de parada es por número de iteraciones, en lugar de iteraciones consecutivas sin actualizar el incumbente.*

Tabla 5.5: Comparación de la calidad de las soluciones obtenidas por el algoritmo según distintos criterios de paro

Por lo tanto, el procedimiento metaheurístico híbrido GRASP-VNS que se propone en este trabajo se detiene al alcanzar 500 iteraciones sin mejorar el valor de la función objetivo de la solución incumbente o al llegar a 1000 iteraciones GRASP.

## 5.4 RESULTADOS COMPUTACIONALES

El algoritmo de tipo heurístico GRASP-VNS fue codificado en lenguaje C++ y la experimentación computacional si realizó en el mismo equipo descrito en la Sección 3.2.

El algoritmo fue ejecutado diez veces por instancia y se guardaron la peor y la

mejor solución, así como el tiempo de cómputo que fue necesario para obtener esta última.

Clase	Mejor gap con respecto al óptimo( %)	Peor gap con respecto al óptimo( %)	Tiempo de ejecución	Gap entre la mejor y la peor solución (%)
1	0.000	0.492	149.162	1.526
2	0.606	0.764	100.104	1.473
3	0.000	1.458	41.918	2.170
4	0.114	0.893	288.064	1.788
5	0.015	0.349	323.699	2.375
6	0.523	1.619	233.807	1.539

Tabla 5.6: Resultados del algoritmo híbrido GRASP-VNS

En la Tabla 5.6 se reportan el mejor y el peor gap con respecto al óptimo para las soluciones encontradas por instancia en las diez ejecuciones del algoritmo, así como el gap entre estos valores. En el apéndice A se muestran los resultados en detalle por instancia.

Nótese que en la clase 1 y en la 3, el híbrido GRASP-VNS encontró la solución óptima en todas las instancias para las cuales se conoce dicho valor. Además, tanto en el mejor como en el peor caso, dicho promedio no excedió al 2% en todas las clases. En las figuras 5.2 y 5.3 se muestran los intervalos de confianza para los gaps en el mejor y en el peor caso, respectivamente. Con una confiabilidad del 95% podemos afirmar que en el mejor caso el peor gap promedio puede ser de menos de 2% y en el peor caso este valor no puede exceder al 4%. Nótese que algunos de los intervalos toman valores negativos, pero estos valores no se consideran ya que la heurística que se propone en esta tesis proporciona es primal, por lo cual proporciona cotas inferiores del óptimo.

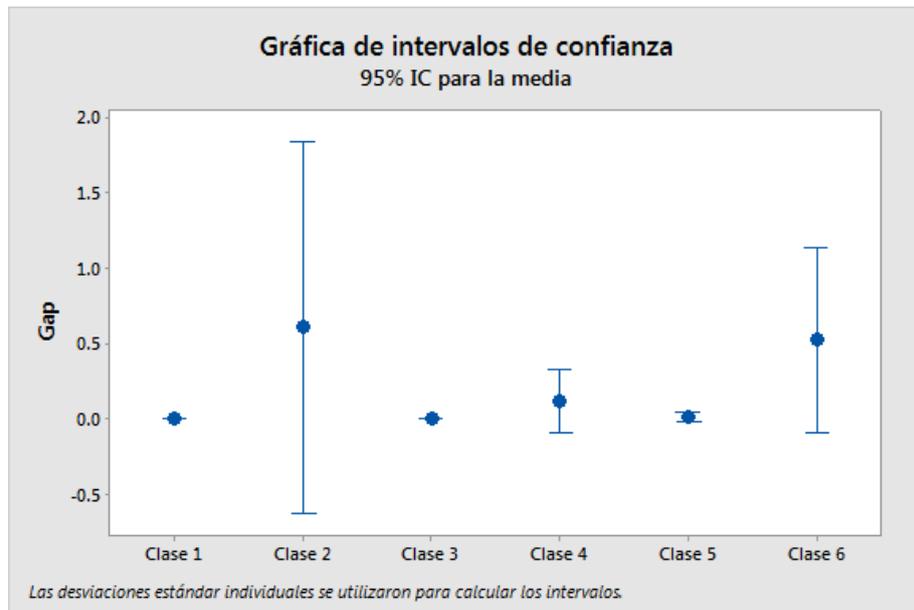


Figura 5.2: Intervalos de confianza para la media del mejor gap en cada clase con una confiabilidad del 95 %

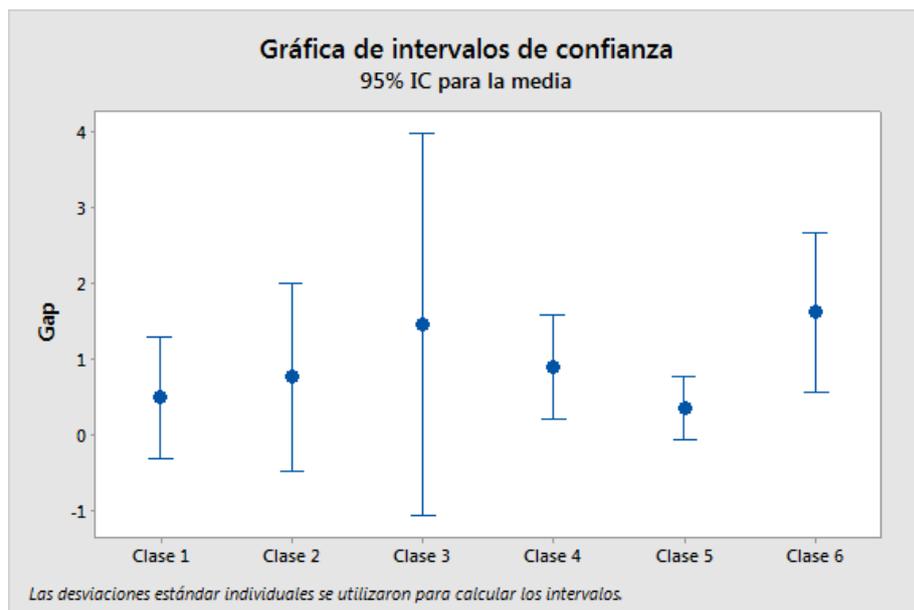


Figura 5.3: Intervalos de confianza para la media del peor gap en cada clase con una confiabilidad del 95 %

Por otro lado, se puede notar que no existe una gran variación entre los mejores y los peores resultados, al haber un gap máximo de 2.375 % entre dichos valores. Por último, con respecto al tiempo de ejecución, nótese que este es mucho menor en la

clase 3, en la cual los grafos tienen el 30 % de los nodos obligatorios y el nivel de conflicto es alto. Como consecuencia, en las instancias de dicha clase el espacio de búsqueda es más reducido que en el resto de las clases, por lo cual, el algoritmo converge más rápidamente.

Para comprobar de manera experimental el resultado anterior se hizo una prueba de Kruskal Wallis bajo la hipótesis nula de que no existen diferencias significativas entre la mediana de los tiempos de ejecución del híbrido GRASP-VNS en cada una de las clases contra la hipótesis nula de que, al menos en una clase, el tiempo de cómputo difiere con respecto a las otras. Dichas hipótesis se representan por las ecuaciones (5.5) y (5.6).

<b>Version GRASP-VNS</b>	<b>N</b>	<b>Mediana</b>	<b>Clasificación del promedio</b>	<b>Z</b>
1	62	15.310	173.1	0.23
2	55	10.290	150.8	-1.62
3	53	3.920	116.6	-4.34
4	62	25.710	196.3	2.28
5	55	28.970	195.5	2.06
6	53	26.010	185.6	1.22
General	340		170.5	

H = 27.27    GL = 5    P = 0.000

Tabla 5.7: Prueba de Kruskal Wallis: Tiempo de ejecución contra clase de instancias

De la Tabla 5.7 se puede concluir con un valor  $p$  despreciable que se rechaza la hipótesis nula en favor de la alternativa. Por lo cual, existe al menos una clase de instancias para la cual el tiempo requerido por el algoritmo metaheurístico para obtener una solución es diferente al tiempo que es necesario en el resto de las clases.

Con el fin de determinar cuál o cuáles de las clases de instancias son las que difieren en cuanto al tiempo de ejecución promedio, se obtuvieron los intervalos de confianza para la media de dichos tiempos, aplicando la corrección de Bonferroni, tal como se muestra en la Tabla 5.8 y en la Figura 5.4.

Clase	Límite inferior	Límite superior
1	55.0358	243.287
2	33.2986	166.909
3	4.88128	78.9553
4	98.8923	477.236
5	43.3755	604.023
6	87.1367	380.477

Tabla 5.8: Intervalos de confianza para la media del gap de cada versión del GRASP-VNS con una confiabilidad del 95 %

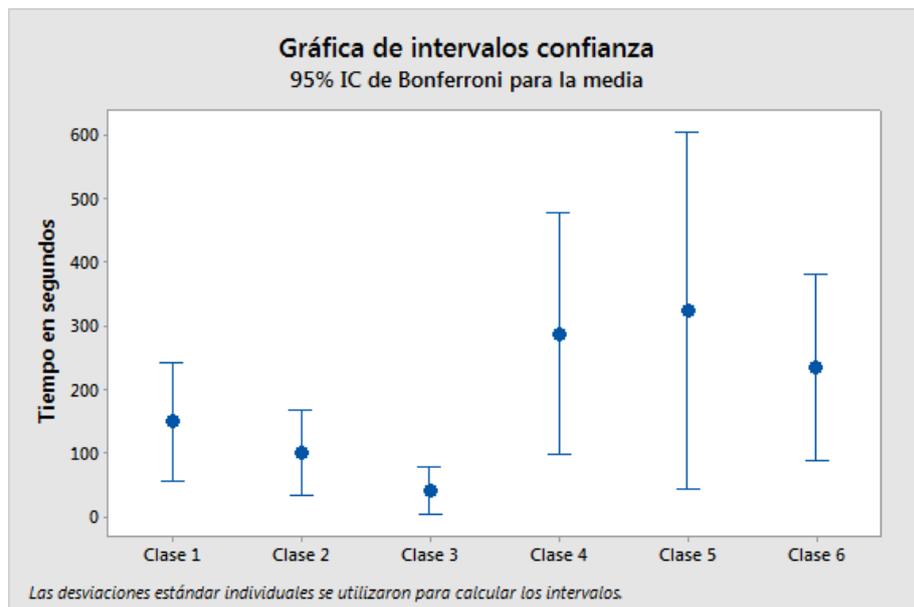


Figura 5.4: Intervalos de confianza para la media del tiempo de ejecución en cada clase con una confiabilidad del 95 %

De la Tabla 5.8 y la Figura 5.4 se puede concluir que el tiempo requerido para la ejecución del algoritmo en las instancias de la clase 3 es menor que el tiempo necesario para las instancias de las clases 4 y 5. Sin embargo, con respecto al resto de las clases no se pueden establecer conclusiones precisas con esta prueba.

En la Tabla 5.9 se establecen comparaciones entre las soluciones obtenidas por el algoritmo heurístico GRASP-VNS y las mejores soluciones conocidas al resolver las instancias de la Clase 1 con los modelos M1 y M2, mediante el uso de Cplex. Para cada instancia se reporta, en la segunda columna de la tabla, la mejor solución

encontrada por el heurístico en 10 corridas de experimentación. La tercera columna muestra el mejor resultado conocido que fue obtenido a través de los modelos M1 y M2 y la cuarta, el gap de la solución del heurístico con respecto a este. En las columnas quinta y sexta se reportan los tiempos de cómputo requeridos para obtener las soluciones de las columnas dos y tres, respectivamente.

Instancia	GRASP-VNS	Mejor conocida	Gap (%)	GRASP-VNS (seg)	Mejor conocida (seg)
att48.5314	15	15	0.00	5.98	73.96
att48.7971	17	17	0.00	5.59	11.18
att48.4298	9	9	0.00	2.14	254.55
att48.4688	13	13	0.00	4.31	43.42
att48.5079	14	14	0.00	5.78	66.16
cmt121.409	535	-	-	92.4783	>3600
cmt121.332	416	-	-	121.79	>3600
cmt121.362	501	-	-	168.62	>3600
cmt121.393	530	-	-	113.09	>3600
cmt151.350	809	-	-	398.04	>3600
cmt151.525	872	872	0.00	76.93	889.38
cmt151.241	434	-	-	240.7	>3600
cmt151.263	558	-	-	366.34	>3600
cmt151.285	664	-	-	306.7	>3600
cmt200.282	595	-	-	506.73	>3600
cmt200.573	1352	-	-	236.59	>3600
cmt200.295	745	-	-	489.88	>3600
cmt200.322	891	-	-	628.11	>3600
cmt200.348	1001	-	-	673.55	>3600
eil30.286	6375	6375	0.00	5.03	2.14
eil30.221	5125	5125	0.00	2.72	50.74
eil30.241	5775	5775	0.00	5.46	19.37
eil30.261	6275	6275	0.00	5.81	10.47
eil33.221	5230	5230	0.00	5.94	120.51
eil33.331	14380	14380	0.00	7.83	7.76
eil33.233	7430	7430	0.00	7.61	99.07
eil33.254	11630	11630	0.00	9.19	19.94
eil33.276	12830	12830	0.00	9.59	29.88

La tabla continúa en la página siguiente

Continuación de la página anterior					
Instancia	GRASP-VNS	Mejor conocida	Gap (%)	GRASP-VNS (seg)	Mejor conocida (seg)
eil51.213	245	245	0.00	18.24	108.90
eil51.320	287	287	0.00	11.35	6.41
eil51.143	122	122	0.00	6.09	566.98
eil51.156	150	150	0.00	12.98	118.98
eil51.169	177	177	0.00	17.64	351.38
eil76.269	520	520	0.00	39.56	2206.36
eil76.404	599	599	0.00	21.45	24.66
eil76.175	232	232	0.00	25.92	2801.55
eil76.191	312	312	0.00	33.18	2608.90
eil76.207	367	364	-0.82	35.2	>3600
eil101.315	565	-	-	83.8	>3600
eil101.472	612	612	0.00	31.83	244.971
eil101.188	281	-	-	47.01	>3600
eil101.205	367	-	-	64.68	>3600
eil101.222	414	-	-	70.25	>3600
gil262.1189	4103	-	-	1324.14	>3600
gil262.1784	4980	-	-	440.55	>3600
gil262.859	2502	-	-	832.27	>3600
gil262.937	3024	-	-	883.44	>3600
gil262.1015	3555	-	-	722.83	>3600
op21.3449	165	165	0.00	1.06	0.35
op21.2703	135	135	0.00	0.34	0.35
op21.2973	150	150	0.00	0.79	0.66
op21.3244	155	155	0.00	0.99	0.75
op32.4127	85	85	0.00	3.48	5.69
op32.6191	115	115	0.00	2.55	0.18
op32.3105	35	35	0.00	0.29	3.74
op32.3415	60	60	0.00	1.27	4.24
op32.3726	75	75	0.00	2.76	5.73
op33.4878	260	260	0.00	3.7	3.58
op33.7317	330	330	0.00	2.97	1.22
op33.2905	110	110	0.00	0.45	1.62
op33.3196	160	160	0.00	0.88	4.66
op33.3486	180	180	0.00	1.55	0.61
<b>Soluciones óptimas encontradas por GRASP-VNS: 39 de 39 conocidas</b>					
<b>Suma de tiempos de solución GRASP-VNS: 9,248.02 seg.</b>					
<b>Suma de tiempos de solución Cplex: 93,571 seg. (incluyendo las instancias que tomaron 3600 seg.)</b>					

Tabla 5.9: Comparaciones entre los resultados obtenidos mediante la heurística GRASP-VNS y las mejores soluciones conocidas de los modelos M1 y M2 en la Clase 1

En los casos en los que se indica un tiempo  $>3600$  es porque el solver consumió la totalidad del tiempo de solución asignado, asimismo en los casos en los que no se reporta una solución obtenida por Cplex es porque, dentro del tiempo límite, este no fue capaz de encontrar una solución entera al problema y, en consecuencia, no es posible calcular un gap. Por último, un gap negativo indica que la heurística obtuvo una solución cuyo valor de la función objetivo es más alto que el de la mejor solución encontrada por Cplex en 3600 segundos, en tal caso la solución reportada por el solver no sería óptima, sino entera factible.

De la tabla anterior se puede notar que el algoritmo híbrido GRASP-VNS fue capaz de encontrar todas las soluciones óptimas conocidas, además de mejorar una solución entera reportada por Cplex en un 0.82 %.

Nótese que en algunas instancias de 30 nodos o menos, el tiempo de cómputo requerido por la heurística es mayor que el tiempo requerido por el solver; sin embargo, en general se hace notable la ventaja del uso del algoritmo GRASP-VNS ya que, mientras que en 93571 segundos (casi 26 horas) el solver solo pudo resolver 40 de 62 instancias, la heurística ofrece soluciones para todos los casos en tan solo el 9.88 % de dicho tiempo (poco más de 2 horas y media).

La Tabla 5.10 muestra comparaciones entre las mejores soluciones encontradas por el GRASP-VNS y las mejores soluciones conocidas al resolver los modelos M1 y M2 mediante Cplex en las instancias de la Clase 2 y sigue el mismo formato de la Tabla 5.9.

Instancia	GRASP-VNS	Mejor conocida	Gap (%)	GRASP-VNS (seg)	Mejor conocida (seg)
att48.7971	12	12	0	1.46	20.214
att48.6994	11	11	0	2.79	43.836
att48.7630	12	12	0	1.52	36.654
att48.8265	12	12	0	1.46	16.284
cmt121.409	330	-	-	116.98	>3600
cmt121.374	305	-	-	196.69	>3600
cmt121.408	329	-	-	130.26	>3600
cmt121.442	330	-	-	52.11	>3600
cmt151.350	460	-	-	108.41	>3600
cmt151.525	561	-	-	46.26	>3600
cmt151.338	403	-	-	203.12	>3600
cmt151.368	496	-	-	211.32	>3600
cmt151.399	534	-	-	173.95	>3600
cmt200.573	825	-	-	147.87	>3600
cmt200.376	481	-	-	449.16	>3600
cmt200.410	591	-	-	432.96	>3600
cmt200.445	694	-	-	460.67	>3600
eil30.286	2925	2925	0	1.39	3.847
eil30.226	1625	1625	0	0.59	3.599
eil30.246	1625	1625	0	0.61	5.92
eil30.266	2475	2475	0	0.82	5.727
eil33.331	8480	8480	0	3.71	4.985
eil33.319	8480	8480	0	3.95	2.52
eil33.348	8480	8480	0	3.7	13.946
eil33.377	9430	9430	0	3.31	3.332
eil51.320	168	168	0	6.42	2.824
eil51.245	148	148	0	11.42	134.482
eil51.268	168	168	0	10.29	17.72
eil51.290	168	168	0	8.18	2.26
eil76.269	377	354	-6.4971751412	22.67	>3600
eil76.404	409	409	0	9.65	5.332
eil76.209	201	201	0	26.79	593.724
eil76.228	294	294	0	21.39	798.089
eil76.247	345	345	0	25.56	1135.908
eil101.315	383	-	-	51.21	>3600
eil101.472	409	409	0	21.4	138.441
eil101.255	260	-	-	100.57	>3600
eil101.278	326	-	-	56.26	>3600
eil101.302	370	-	-	109.54	>3600
gil262.1784	3003	-	-	213.06	>3600
gil262.1327	2038	-	-	768.49	>3600
gil262.1447	2759	-	-	694.05	>3600
gil262.1568	2956	-	-	586.31	>3600

La tabla continúa en la página siguiente

Continuación de la página anterior					
Instancia	GRASP-VNS	Mejor conocida	Gap (%)	GRASP-VNS (seg)	Mejor conocida (seg)
op21_2289	60	60	0	0.09	0.209
op21_3449	125	125	0	0.18	0.128
op21_2305	60	60	0	0.09	0.221
op21_2536	60	75	20	0.09	0.305
op21_2766	80	80	0	0.12	0.413
op32_6191	80	80	0	0.81	0.565
op32_5399	70	70	0	1.21	1.292
op32_5890	80	80	0	1	1.879
op33_7317	210	210	0	1.04	1.169
op33_6808	170	170	0	0.86	1.539
op33_7489	210	210	0	1.02	1.029
op33_8170	220	220	0	0.88	0.231
<b>Soluciones óptimas encontradas por GRASP-VNS: 32 de 33 conocidas</b>					
<b>Suma de tiempos de solución GRASP-VNS: 5,505.72 seg.</b>					
<b>Suma de tiempos de solución Cplex: 82,198.62 seg. (incluyendo las instancias que tomaron 3600 seg.)</b>					

Tabla 5.10: Comparaciones entre los resultados obtenidos mediante la heurística GRASP-VNS y las mejores soluciones conocidas de los modelos M1 y M2 en la Clase 2

En la tabla anterior se puede notar que el algoritmo GRASP-VNS fue capaz de resolver las 55 instancias de la Clase 2 en un tiempo máximo de 768.49 segundos, además de ser capaz de encontrar 32 soluciones óptimas de 33 conocidas. Si bien, la metaheurística no encontró una de las soluciones óptimas, fue capaz de mejorar una solución entera factible encontrada por Cplex.

Ahora bien, cabe destacar que en un total de 82198.62 segundos (casi 23 horas) Cplex pudo encontrar soluciones no necesariamente óptimas para 34 de 55 instancias pertenecientes a la Clase 2, mientras que el híbrido GRASP-VNS ofrece soluciones para todos los casos en tan solo el 6.7% de ese tiempo (poco más de una hora y media).

La Tabla 5.11 muestra información análoga a las dos tablas anteriores para las instancias pertenecientes a la Clase 3.

Instancia	GRASP-VNS	Mejor conocida	Gap (%)	GRASP-VNS (seg)	Mejor conocida (seg)
att48.5314	4	4	0	2.09	37.167
att48.7971	7	7	0	1.47	3.497
att48.5389	4	4	0	0.4	40.052
att48.5879	5	5	0	2.57	31.17
att48.6369	5	5	0	2.18	19.806
cmt121.409	152	-	-	43.26	>3600
cmt121.395	152	-	-	47.6	>3600
cmt121.431	152	-	-	39.13	>3600
cmt121.467	152	-	-	28.74	>3600
cmt151.525	199	-	-	24.66	>3600
cmt151.433	171	-	-	87.21	>3600
cmt151.473	199	-	-	37.11	>3600
cmt151.512	199	-	-	26.69	>3600
cmt200.573	463	-	-	65.98	>3600
cmt200.469	354	-	-	218.79	>3600
cmt200.511	438	-	-	179.75	>3600
cmt200.554	463	-	-	78.9	>3600
eil30.286	950	950	0	0.67	2.012
eil30.310	1950	1950	0	0.6	2.025
eil30.338	2075	2075	0	0.64	0.293
eil30.367	2075	2075	0	0.55	0.125
eil33.371	5530	5530	0	0.82	1.172
eil33.404	5530	5530	0	0.61	1.416
eil33.438	5530	5530	0	0.57	0.086
eil51.320	99	99	0	2.67	1.811
eil51.277	90	90	0	4.87	77.196
eil51.302	99	99	0	3.92	6.757
eil51.328	99	99	0	2.56	3.902
eil76.269	167	167	0	24.31	37.087
eil76.404	202	202	0	4.24	5.833
eil76.264	142	142	0	18.86	142.45
eil76.288	193	183	-5.4644808743	15.2	>3600
eil76.312	202	202	0	7.83	34.737
eil101.472	175	175	0	7.96	466.002
eil101.352	155	-	-	40.25	>3600
eil101.384	175	-	-	15.75	>3600
eil101.416	175	175	0	11.05	1175.793
gil262.1784	1369	-	-	115.73	>3600
gil262.1481	1147	-	-	578.91	>3600
gil262.1615	1366	-	-	346.03	>3600
gil262.1750	1369	-	-	128.05	>3600
op21.3349	45	45	0	0.04	0.029
op21.2707	30	30	0	0.04	0.064
op21.2978	45	45	0	0.04	0.042
op21.3519	45	45	0	0.04	0.031

La tabla continúa en la página siguiente

Continuación de la página anterior					
Instancia	GRASP-VNS	Mejor conocida	Gap (%)	GRASP-VNS (seg)	Mejor conocida (seg)
op32_6191	45	45	0	0.04	0.458
op32_5511	45	45	0	0.51	0.325
op32_6062	45	45	0	0.54	0.184
op32_6613	45	45	0	0.38	0.438
op33_7317	130	130	0	0.18	0.063
op33_5511	80	80	0	0.21	0.4
op33_6062	120	120	0	0.24	1.056
op33_6613	130	130	0	0.23	0.111
<b>Soluciones óptimas encontradas por GRASP-VNS: 34 de 34 conocidas</b>					
<b>Suma de tiempos de solución GRASP-VNS: 2,221.67 seg.</b>					
<b>Suma de tiempos de solución Cplex: 70,493.59 seg. (incluyendo las instancias que tomaron 3600 seg.)</b>					

Tabla 5.11: Comparaciones entre los resultados obtenidos mediante la heurística GRASP-VNS y las mejores soluciones conocidas de los modelos M1 y M2 en la Clase 3

En la Tabla 5.11 se puede apreciar que en un tiempo de 2221.67 segundos (poco más de 37 minutos) el algoritmo GRASP-VNS puede ofrecer soluciones para las 53 instancias de la clase 3, mientras que a Cplex le tomó 70493.59 segundos (poco más de 19 horas y media) resolver tan solo 34 casos. En cuanto a la calidad de las soluciones, la metaheurística logró resolver a optimalidad las 34 instancias para las cuales la solución óptima es conocida y logró mejorar en un 5.46% una solución entera factible encontrada por el solver.

Resultados análogos a los reportados en las Tablas 5.9 a 5.11 para la Clase 4 se pueden encontrar en la Tabla 5.12.

Instancia	GRASP-VNS	Mejor conocida	Gap (%)	GRASP-VNS (seg)	Mejor conocida (seg)
att48.5314	21	21	0.00	7.56	133.13
att48.7971	28	29	3.45	9.74	474.22
att48.4298	13	13	0.00	3.45	250.48
att48.4688	17	17	0.00	6.09	145.80
att48.5079	20	20	0.00	7.42	108.12
cmt121.409	803	-	-	465.44	>3600
cmt121.332	512	-	-	257.09	>3600
cmt121.362	644	-	-	412.98	>3600
cmt121.393	751	-	-	381.74	>3600
cmt151.350	1048	-	-	620.91	>3600
cmt151.525	1446	-	-	349.43	>3600
cmt151.241	469	-	-	306.97	>3600
cmt151.263	629	-	-	326.59	>3600
cmt151.285	769	-	-	358.02	>3600
cmt200.282	723	-	-	594.27	>3600
cmt200.573	2097	-	-	1770.94	>3600
cmt200.295	869	-	-	646.46	>3600
cmt200.322	1066	-	-	802.44	>3600
cmt200.348	1237	-	-	662.52	>3600
eil30.286	9375	9375	0.00	10.02	125.55
eil30.221	5450	5450	0.00	3.54	163.53
eil30.241	7300	7300	0.00	8.93	23.76
eil30.261	8575	8575	0.00	8.34	33.19
eil33.221	6950	6950	0.00	7.39	120.36
eil33.331	19730	19730	0.00	14.09	79.45
eil33.233	10630	10630	0.00	10.88	25.41
eil33.254	14180	14180	0.00	11.58	51.88
eil33.276	17430	17430	0.00	16.01	15.23
eil51.213	328	328	0.00	47.64	755.24
eil51.320	509	509	0.00	30.12	611.35
eil51.143	168	168	0.00	9.62	751.52
eil51.156	201	201	0.00	20.03	420.81
eil51.169	232	232	0.00	21.3	868.37
eil76.269	616	514	-19.84	118	>3600
eil76.404	919	923	0.43	130.42	3314.44
eil76.175	274	269	-1.86	35.15	>3600
eil76.191	366	311	-17.68	46.38	>3600
eil76.207	442	-	-	63.91	>3600
eil101.315	774	730	-6.03	233.44	>3600
eil101.472	1057	-	-	221.92	>3600
eil101.188	326	-	-	84.38	>3600
eil101.205	422	-	-	150.17	>3600
eil101.222	489	-	-	108.25	>3600

La tabla continúa en la página siguiente

Instancia	GRASP-VNS	Mejor conocida	Gap (%)	GRASP-VNS (seg)	Mejor conocida (seg)
gil262_1189	5276	-	-	1974.23	>3600
gil262_1784	7665	-	-	2931.03	>3600
gil262_859	2842	-	-	976.81	>3600
gil262_937	3714	-	-	1087.86	>3600
gil262_1015	4062	-	-	1450.23	>3600
op21_3449	260	260	0.00	2.63	5.95
op21_2703	180	180	0.00	1.08	1.31
op21_2973	215	215	0.00	2.3	1.88
op21_3244	260	260	0.00	2.44	1.50
op32_4127	110	110	0.00	3.93	6.79
op32_6191	180	180	0.00	3.23	4.61
op32_3105	35	35	0.00	0.3	3.29
op32_3415	75	75	0.00	1.65	2.70
op32_3726	90	90	0.00	2.78	3.26
op33_4878	310	310	0.00	6.54	6.06
op33_7317	460	460	0.00	6.46	5.69
op33_2905	120	120	0.00	0.61	5.78
op33_3196	180	180	0.00	1.45	6.73
op33_3486	220	220	0.00	2.86	4.33
<b>Soluciones óptimas encontradas por GRASP-VNS: 32 de 34 conocidas</b>					
<b>Suma de tiempos de solución GRASP-VNS: 17,859.99 seg.</b>					
<b>Suma de tiempos de solución Cplex: 109,331.69 seg. (incluyendo las instancias que tomaron 3600 seg.)</b>					

Tabla 5.12: Comparaciones entre los resultados obtenidos mediante la heurística GRASP-VNS y las mejores soluciones conocidas de los modelos M1 y M2 en la Clase 4

De la tabla anterior se puede concluir que el algoritmo heurístico GRASP-VNS fue capaz de resolver a optimalidad 32 de las 34 instancias para las cuales se conoce el óptimo, en los dos casos que no resolvió a optimalidad el gap máximo fue de 3.45% y, además, logró mejorar hasta en un 19.84% cuatro soluciones enteras factibles encontradas por Cplex.

En cuanto al tiempo de ejecución, a la metaheurística le tomó 17859.99 segundos (casi 5 horas) encontrar soluciones para las 62 instancias que conforman la Clase 4, mientras que en 109331.69 (poco más de 30 horas y media) Cplex solo encontró soluciones para 38 casos.

De manera similar a las clases de instancias anteriores, en la Tabla 5.13 se muestra la mejor solución encontrada por la metaheurística en 10 corridas de experimentación y la mejor solución reportadas por Cplex para cada instancia de la Clase 5.

Instancia	GRASP-VNS	Mejor conocida	Gap (%)	GRASP-VNS (seg)	Mejor conocida (seg)
att48.7971	24	24	0.00	6.37	184.50
att48.6994	19	19	0.00	5.88	414.64
att48.7630	23	23	0.00	6.64	184.72
att48.8265	25	25	0.00	6.41	174.80
cmt121.409	621	-	-	336.34	>3600
cmt121.374	444	-	-	368.46	>3600
cmt121.408	615	-	-	409.95	>3600
cmt121.442	685	-	-	240.98	>3600
cmt151.350	657	-	-	359.49	>3600
cmt151.525	1168	-	-	467.02	>3600
cmt151.338	584	-	-	368.19	>3600
cmt151.368	735	-	-	280.49	>3600
cmt151.399	861	-	-	470.70	>3600
cmt200.573	1615	-	-	858.56	>3600
cmt200.376	636	-	-	880.79	>3600
cmt200.410	901	-	-	737.14	>3600
cmt200.445	1141	-	-	693.69	>3600
eil30.286	6225	6225	0.00	8.52	59.38
eil30.226	3350	3350	0.00	3.51	19.09
eil30.246	4750	4750	0.00	3.86	19.72
eil30.266	5075	5075	0.00	7.64	93.20
eil33.331	14230	14230	0.00	9.97	47.85
eil33.319	12880	12880	0.00	9.40	65.60
eil33.348	14930	14930	0.00	9.81	7.88
eil33.377	15680	15680	0.00	7.52	82.88
eil51.320	402	402	0.00	21.73	277.59
eil51.245	235	236	0.42	21.71	1027.57
eil51.268	302	302	0.00	27.02	1252.08
eil51.290	357	357	0.00	28.97	840.48

La tabla continúa en la página siguiente

Continuación de la página anterior					
Instancia	GRASP-VNS	Mejor conocida	Gap (%)	GRASP-VNS (seg)	Mejor conocida (seg)
eil76_269	481	-	-	56.98	>3600
eil76_404	745	747	0.27	52.06	>3600
eil76_209	228	-	-	49.49	>3600
eil76_228	341	-	-	62.67	>3600
eil76_247	411	-	-	76.48	>3600
eil101_315	546	529	-3.21	171.98	>3600
eil101_472	854	855	0.12	178.14	>3600
eil101_255	296	-	-	133.14	>3600
eil101_278	409	-	-	110.45	>3600
eil101_302	485	-	-	160.06	>3600
gil262_1784	6079	-	-	1913.11	>3600
gil262_1327	2974	-	-	1566.06	>3600
gil262_1447	4244	-	-	1679.43	>3600
gil262_1568	5039	-	-	4899.67	>3600
op21_2289	135	135	0.00	0.17	0.24
op21_3449	235	235	0.00	1.09	2.52
op21_2305	135	135	0.00	0.18	0.28
op21_2536	135	135	0.00	0.17	0.45
op21_2766	165	165	0.00	0.27	2.84
op32_6191	150	150	0.00	3.05	7.20
op32_5399	110	110	0.00	2.65	8.08
op32_5890	140	140	0.00	3.05	10.91
op33_7317	400	400	0.00	8.35	0.95
op33_6808	270	270	0.00	2.85	26.18
op33_7489	410	410	0.00	7.91	20.34
op33_8170	440	440	0.00	7.25	6.25
<b>Soluciones óptimas encontradas por GRASP-VNS: 27 de 28 conocidas</b>					
<b>Suma de tiempos de solución GRASP-VNS: 17,803.47 seg.</b>					
<b>Suma de tiempos de solución Cplex: 102,038.21 seg. (incluyendo las instancias que tomaron 3600 seg.)</b>					

Tabla 5.13: Comparaciones entre los resultados obtenidos mediante la heurística GRASP-VNS y las mejores soluciones conocidas de los modelos M1 y M2 en la Clase 5

De la tabla anterior se puede notar que en 17803.47 segundos (casi 5 horas) el algoritmo heurístico encontró soluciones para las 55 instancias pertenecientes a la Clase 5, mientras que el solver resolvió a optimalidad 28 instancias y encontró soluciones enteras factibles para 3 en 102038.21 segundos (poco más de 28 horas y

media). En cuanto a la calidad de las soluciones obtenidas por el híbrido GRASP-VNS, este fue capaz de resolver a optimalidad 27 instancias de las 28 para las cuales el óptimo es conocido y obtuvo un gap de 0.42% en la instancia restante. Además, mejoró una de las soluciones enteras en un 3.21%, y en las otras dos obtuvo gaps menores al 0.3%.

Finalmente, en la Tabla 5.14 se muestran comparaciones entre las mejores soluciones obtenidas por la metaheurística y la mejor solución conocida al resolver las instancias de la Clase 6 con los modelos M1 y M2 usando el solver Cplex.

Instancia	GRASP-VNS	Mejor conocida	Gap (%)	GRASP-VNS (seg)	Mejor conocida (seg)
att48.5314	11	11	0.00	5.35	225.67
att48.7971	19	19	0.00	8.7	735.09
att48.5389	12	12	0.00	4.9	181.00
att48.5879	14	14	0.00	6.46	495.58
att48.6369	15	16	6.25	7.51	284.14
cmt121.409	473	-	-	348.92	>3600
cmt121.395	429	-	-	360.05	>3600
cmt121.431	540	-	-	379.92	>3600
cmt121.467	608	-	-	263	>3600
cmt151.525	838	-	-	465.13	>3600
cmt151.433	488	-	-	413.18	>3600
cmt151.473	702	-	-	458.7	>3600
cmt151.512	824	-	-	465.22	>3600
cmt200.573	1358	-	-	639.65	>3600
cmt200.469	792	-	-	773.02	>3600
cmt200.511	1147	-	-	791.76	>3600
cmt200.554	1327	-	-	801.21	>3600
eil30.286	1550	1550	0.00	4.07	31.86
eil30.310	4775	4775	0.00	4.04	28.53
eil30.338	5550	5550	0.00	2.87	15.69
eil30.367	5850	5850	0.00	2.08	18.23
eil33.371	10600	10600	0.00	5.37	32.38
eil33.404	13330	13330	0.00	4.48	61.31
eil33.438	14030	14030	0.00	2.18	27.29

La tabla continúa en la página siguiente

Continuación de la página anterior					
Instancia	GRASP-VNS	Mejor conocida	Gap (%)	GRASP-VNS (seg)	Mejor conocida (seg)
eil51_320	320	329	2.74	29.32	175.54
eil51_277	212	212	0.00	18.08	136.53
eil51_302	272	289	5.88	23.4	221.04
eil51_328	343	344	0.29	26.01	267.91
eil76_269	229	228	-0.44	59.52	>3600
eil76_404	599	-	-	37.94	>3600
eil76_264	211	211	0.00	60.5	3134.23
eil76_288	304	-	-	51.58	>3600
eil76_312	380	-	-	47.48	>3600
eil101_472	614	631	2.69	163.29	>3600
eil101_352	337	-	-	155.74	>3600
eil101_384	460	-	-	161.93	>3600
eil101_416	552	-	-	166.03	>3600
gil262_1784	4444	-	-	944.43	>3600
gil262_1481	2493	-	-	1862.28	>3600
gil262_1615	3660	-	-	1124.5	>3600
gil262_1750	4352	-	-	1215.46	>3600
op21_3349	155	155	0.00	0.76	0.91
op21_2707	75	75	0.00	0.21	0.35
op21_2978	110	110	0.00	0.5	0.54
op21_3519	160	160	0.00	0.9	1.07
op32_6191	115	115	0.00	3.1	1.24
op32_5511	65	65	0.00	3.31	4.30
op32_6062	105	105	0.00	2.97	3.86
op32_6613	130	130	0.00	2.72	5.16
op33_7317	350	350	0.00	2.82	0.89
op33_5511	130	130	0.00	2.6	5.68
op33_6062	240	240	0.00	3.24	5.56
op33_6613	300	300	0.00	3.37	3.18
<b>Soluciones óptimas encontradas por GRASP-VNS: 25 de 29 conocidas</b>					
<b>Suma de tiempos de solución GRASP-VNS: 12,391.76 seg.</b>					
<b>Suma de tiempos de solución Cplex: 92,504.752 seg. (incluyendo las instancias que tomaron 3600 seg.)</b>					

Tabla 5.14: Comparaciones entre los resultados obtenidos mediante la heurística GRASP-VNS y las mejores soluciones conocidas de los modelos M1 y M2 en la Clase 6

En la tabla anterior se puede observar que solamente 29 de las 53 instancias de la Clase 6 pudieron ser resueltas a optimalidad por Cplex en 92504.752 segundos

(más de 25 horas y media) y además encontró dos soluciones enteras factibles. Por otro lado, en 12391.76 segundos (casi 4 horas) el híbrido GRASP-VNS encontró 25 óptimos de los 29 que se conocen, obteniendo un gap máximo de 6.25 % en las 4 instancias restantes y, en las dos instancias para las cuales Cplex encontró una solución entera factible obtuvo gaps de 2.69 % y -0.44 %.

De las Tablas 5.9 a 5.14 se puede concluir que haciendo uso del solver hay muchas instancias para las cuales no se tiene información alguna de la solución de las mismas; mientras que haciendo uso del algoritmo heurístico, si bien no se puede asegurar la optimalidad de las soluciones encontradas, se puede ofrecer una cota inferior para el valor óptimo de la función objetivo en tiempos de cómputo mucho menores que los requeridos por Cplex.

## CAPÍTULO 6

# PROCEDIMIENTO DE GENERACIÓN DE COLUMNAS PROPUESTO

---

En este capítulo se presenta un modelo matemático alternativo a los expuestos en el Capítulo 2 para el problema de estudio que está basado en una formulación de gran escala y que es resuelto a través de una técnica de generación de columnas.

El método de generación de columnas es comúnmente utilizado en problemas cuyo modelo matemático posee un número mucho mayor de variables (columnas) que de restricciones (renglones) y se basa en la idea de que, al resolver el problema con el método símplex, la mayoría de las columnas nunca forman parte de la base, por lo que generarlas y operar sobre ellas implica un esfuerzo computacional innecesario.

La idea principal consiste en formular un modelo equivalente llamado *problema maestro*, el cual invoca de manera iterativa a un *subproblema*. El problema maestro envía variables duales al subproblema que este utiliza para resolver el problema de *pricing*, posteriormente el subproblema responde mandando nuevas columnas que se integrarán al problema maestro. El procedimiento se detiene cuando el subproblema no encuentra nuevas columnas para el maestro.

En este capítulo se presentará una breve introducción al esquema de generación de columnas y se justificará el uso de este algoritmo para resolver el problema propuesto en este trabajo. Además, se exponen resultados computacionales que son comparados contra los que fueron encontrados por el método heurístico.

## 6.1 ASPECTOS TEÓRICOS DEL MÉTODO DE GENERACIÓN DE COLUMNAS

Considere el siguiente problema lineal, donde  $X$  representa el espacio de soluciones definido por un conjunto de restricciones que poseen cierta estructura particular y  $x \in \mathbb{R}^n$ ,  $b \in \mathbb{R}^m$  y  $A \in \mathbb{R}^{m \times n}$

$$\min cx \tag{6.1}$$

sujeto a:

$$Ax = b \tag{6.2}$$

$$x \in X \tag{6.3}$$

Debido a que el problema es lineal, entonces  $X$  es un conjunto convexo y si suponemos que este es no vacío y acotado, entonces todo punto  $x \in X$  se puede representar por la combinación convexa de los extremos de  $X$ , denotados por  $x_1, x_2, \dots, x_t$ .

Por lo tanto, el problema anterior puede reescribirse como el siguiente problema, llamado problema maestro.

$$\min \sum_{i=1}^t (cx_i) \lambda_i \tag{6.4}$$

sujeto a:

$$\sum_{i=1}^t (Ax_i) \lambda_i = b \tag{6.5}$$

$$\sum_{i=1}^t \lambda_i = 1 \tag{6.6}$$

$$\lambda_i \geq 0, \quad i = 1, \dots, t \quad (6.7)$$

Suponga que  $\lambda = (\lambda_B, \lambda_N)$  es una solución factible del problema maestro en cierta iteración del método símplex, donde  $\lambda_B$  y  $\lambda_N$  representan a las variables básicas y no básicas respectivamente; y sean  $\alpha$  y  $\beta$  las variables duales asociadas a las restricciones (6.5) y (6.6).

Dichas variables duales son obtenidas por la expresión  $(\alpha, \beta) = \hat{c}_B B^{-1}$ , donde  $\hat{c}_B$  representa el vector de coeficientes en la función objetivo de las variables básicas y  $B^{-1}$  es la matriz inversa de la base.

Siguiendo la notación anterior, el costo reducido  $\bar{c}_i$  de una variable no básica  $x_i$  está dado por la ecuación (6.8)

$$\bar{c}_i = cx_i - \alpha^t Ax_i - \beta \quad (6.8)$$

El método símplex se detiene cuando no se puede encontrar una variable no básica con costo reducido negativo; sin embargo, es impráctico calcular la expresión (6.8) para cada variable no básica debido al alto número de variables existentes.

Una alternativa consiste en resolver el subproblema (6.9)-(6.10), lo cual es válido debido a que el conjunto  $X$  es convexo y acotado y, por lo tanto, el óptimo del subproblema se encontrará en uno de sus puntos extremos.

$$\text{mín } (c - \alpha^t A)x - \beta \quad (6.9)$$

$$\text{sujeto a: } x \in X \quad (6.10)$$

Si el valor óptimo de la función objetivo del subproblema es menor que cero, entonces la solución óptima  $x^*$  de este, es añadida al problema maestro y el método

símplex decide cuál de las variables básicas abandonará la base por la prueba del cociente mínimo. Si no, la solución  $\lambda = (\lambda_B, \lambda_N)$  es óptima.

## 6.2 USO DE MÉTODOS EXACTOS EN PROBLEMAS RELACIONADOS

Como se mencionó en la Sección 2.1, en la literatura existen múltiples propuestas de algoritmos exactos para resolver el OP como lo son programación dinámica, *branch and bound*, método de planos cortantes y *branch and cut*.

Sin embargo, ninguno de estos métodos comprende la existencia de conflictos entre nodos ni de visitas obligatorias, motivo por el cual, se realizó una extensión de la revisión bibliográfica con el motivo de detectar el tipo de métodos exactos existentes en la literatura de problemas de transporte, de ruteo de vehículos clásicos y con beneficios que contemplen dichas restricciones.

La Tabla 6.1 describe los métodos exactos que se han utilizado para resolver problemas similares al que se propone en este trabajo, se describe el problema que fue abordado y la dimensión de las instancias que se lograron resolver, siendo  $n$  el número de nodos y  $m$  el número de vehículos disponibles.

Autores	Problema	Método de solución	Tamaño de instancias	
			n	m
Laporte y Martello (1990)	Orienteering Problem	Branch and bound	90	1
Pekny <i>et al.</i> (1990)	Asymmetric Traveling Salesman Problem, Orienteering Problem y Prize Collecting Traveling Salesman Problem	Branch and bound (paralelo)	200	1
Ramesh <i>et al.</i>	Orienteering Problem	Branch and bound + relajación lagrangeana	150	1
Leifer y Rosenwein (1994)	Orienteering Problem	Planos cortantes	33	1
Fischetti <i>et al.</i> (1998)	Orienteering Problem	Branch and cut	500	1
Gendreau <i>et al.</i> (1998a)	Orienteering Problem con nodos obligatorios	Branch and cut	300	1
Butt y Ryan (1999)	Team Orienteering Problem	Generación de columnas	100	10
Sun (2002) <sup>1</sup>	Transportation problem with exclusionary side constraints	Branch and bound	30x40	-
Westerlund <i>et al.</i> (2006)	Traveling Salesman Problem Subtour Problem	Generación de columnas	100	1
Boussier <i>et al.</i> (2007)	Team Orienteering Problem	Branch and Price	102	4
Boussier <i>et al.</i> (2007)	Selective Vehicle Routing Problem with Time Windows	Branch and Price	100	10

La tabla continúa en la página siguiente

<sup>1</sup>Por tratar de un problema de transporte, el grafo es bipartito. 30x40 indica el número de nodos en cada partición. Al no ser un problema de ruteo, no se indica número de vehículos.

Autores	Problema	Método de solución	Tamaño de instancias	
			n	m
Righini y Salani (2009)	Orienteering Problem with Time Windows	Programación dinámica bidireccional con relajación decreciente del espacio de estados	100	1
Bérubé <i>et al.</i> (2009)	Prize Collecting Traveling Salesman Problem	Branch and cut	532	1
Archetti <i>et al.</i> (2013b)	Capacitated Team Orienteering Problem	Branch and price	200	4
Manerba y Mansini (2013) <sup>2</sup>	Multi-Vehicle Traveling Purchaser Problem with Exclusionary Side Constraints	Branch and cut	50	-

Tabla 6.1: Métodos exactos utilizados en problemas afines

Cabe destacar que los métodos más utilizados para resolver problemas afines son de tipo *branch and bound*, *branch and cut*, *branch and price* y generación de columnas, tal como se indica en la Tabla 6.2. Además, cabe destacar que más recientemente ha decrecido el uso del esquema *branch and bound*, ya que los solvers comerciales cuentan con algoritmos de este tipo que son cada vez más sofisticados.

Debido a que los esquemas *branch and price* y generación de columnas han sido ampliamente utilizados y que se siguen utilizando en fechas recientes, se supone que el uso de un algoritmo de generación de columnas es apropiado para encontrar cotas para el problema propuesto en esta tesis.

Además, al integrar problemas tipo OP a un esquema de generación de columnas, usualmente el subproblema es un problema de camino más corto con restriccio-

<sup>2</sup>No se indica un número máximo de vehículos.

Método	Frecuencia	Fecha
Branch and bound	4	1990, 1990, 1992, 2001
Branch and cut	4	1998, 1998, 2009, 2013
Branch and Price	2	2007, 2013
Generación de columnas	2	1999, 2006
Programación dinámica	1	2008
Planos cortantes	1	1994
Relajación lagrangeana	1	1992

Tabla 6.2: Frecuencia y fechas de uso de métodos exactos para problemas afines

nes de recursos, para el cual se conocen algoritmos eficientes para su solución como lo es el algoritmo de programación dinámica de Feillet *et al.* (2004) en conjunto con técnicas de aceleración como programación dinámica bidireccional (Righini y Salani, 2006b) y *decremental state space relaxation* (Righini y Salani, 2008).

## 6.3 FORMULACIÓN MATEMÁTICA DEL OP CON RESTRICCIONES ADICIONALES COMO UN PROBLEMA DE GRAN ESCALA

### 6.3.1 PROBLEMA MAESTRO

Sea  $\Omega$  el conjunto de todas las rutas que comienzan en el nodo 1, terminan en el  $n$  y cuya duración no excede el tiempo límite permitido. Además, se define  $s_r$  como la puntuación generada por la ruta  $r \in \Omega$ ,  $a_{ir}$  igual a 1 si la ruta  $r$  visita el vértice  $i \in V$  y 0 en otro caso,  $\rho_{ir}$  es el número de nodos en el conjunto  $C'_i$ , con  $1 \leq i \leq L$ , que han sido incluidos en la ruta  $r$  y la variable de decisión binaria  $x_r$  igual a 1 si la ruta  $r \in \Omega$  es utilizada y 0 en otro caso. Entonces, el problema maestro está dado por la siguiente formulación.

$$\max z_{PM} = \sum_{r \in \Omega} s_r x_r \quad (6.11)$$

sujeto a:

$$\sum_{r \in \Omega} a_{ir} x_r \leq 1 \quad i \in O \quad (6.12)$$

$$\sum_{r \in \Omega} a_{ir} x_r = 1 \quad i \in M \quad (6.13)$$

$$\sum_{r \in \Omega} \rho_{ir} x_r \leq 1 \quad i = 1, 2, \dots, L \quad (6.14)$$

$$\sum_{r \in \Omega} x_r \leq 1 \quad (6.15)$$

$$x_r \in \{0, 1\} \quad r \in \Omega \quad (6.16)$$

Donde (6.11) maximiza la puntuación de las rutas utilizadas, las restricciones (6.12) aseguran que cada vértice opcional será visitado como máximo una vez y las restricciones (6.13) fuerzan a que cada vértice obligatorio sea visitado en una ocasión. Por último, las restricciones (6.14) aseguran que no se incluyan en la ruta nodos con conflicto entre ellos y la restricción (6.15) asegura que solo se elegirá una ruta.

Cabe destacar que el problema maestro es entero, para poder aplicar la metodología de generación de columnas, la restricción de integralidad es relajada obteniendo un problema maestro relajado, cuya función objetivo será denotada por  $z_{RMP}$ .

### 6.3.2 SUBPROBLEMA

Al resolver el problema maestro por el método símplex se obtienen las variables duales  $\alpha_i, \beta_i$  para cada nodo  $i$  asociadas a las restricciones (6.12), (6.13);  $\gamma_i$  para cada conjunto  $C'_i$  asociadas a las restricciones (6.14); y  $\delta$  relacionada a la restricción (6.15). Entonces, el costo reducido asociado a cada variable está dado por la siguiente ecuación:

$$cr_r = s_r - \sum_{i \in O} a_{ir} \alpha_i - \sum_{i \in M} a_{ir} \beta_i - \sum_{i \in \{1,2,\dots,L\}} \rho_{ir} \gamma_i - \delta \quad (6.17)$$

Sea  $y_i$  una variable binaria que es igual a 1 si una ruta contiene al nodo  $i$  y es 0 en otro caso. Entonces, si  $a_{ir} = 1$  se cumple que  $y_i = 1$  para algún  $i \in V$ . Por lo tanto, la expresión anterior se puede escribir como:

$$cr_r = s_r - \sum_{i \in O} \alpha_i y_i - \sum_{i \in M} \beta_i y_i - \sum_{i \in \{1,2,\dots,L\}} \rho_{ir} \gamma_i - \delta \quad (6.18)$$

Dado que  $s_r$  está definido como la puntuación de los vértices incluidos en la ruta  $r$ , la expresión anterior es equivalente a:

$$cr_r = \sum_{i \in V} s_i y_i - \sum_{i \in O} \alpha_i y_i - \sum_{i \in M} \beta_i y_i - \sum_{i \in \{1,2,\dots,L\}} \rho_{ir} \gamma_i - \delta \quad (6.19)$$

Pero  $s_i = 0$  para  $i \in M$ , por lo tanto:

$$cr_r = \sum_{i \in O} s_i y_i - \sum_{i \in O} \alpha_i y_i - \sum_{i \in M} \beta_i y_i - \sum_{i \in \{1,2,\dots,L\}} \rho_{ir} \gamma_i - \delta \quad (6.20)$$

Agrupando los primeros dos términos se tiene que:

$$cr_r = \sum_{i \in O} (s_i - \alpha_i) y_i - \sum_{i \in M} \beta_i y_i - \sum_{i \in \{1,2,\dots,L\}} \rho_{ir} \gamma_i - \delta \quad (6.21)$$

Pero  $\rho_{ir} = \sum_{j \in C'_i} y_j$ , por lo tanto, la expresión anterior es equivalente a:

$$cr_r = \sum_{i \in O} (s_i - \alpha_i) y_i - \sum_{i \in M} \beta_i y_i - \sum_{i \in \{1,2,\dots,L\}} \left( \gamma_i \sum_{j \in C'_i} y_j \right) - \delta \quad (6.22)$$

De lo anterior, se deduce que la resolución del subproblema es equivalente a

resolver un problema de camino más corto con restricciones de recursos, el cual está dado por el modelo siguiente.

$$\max z_S = \sum_{i \in O} (s_i - \alpha_i) y_i - \sum_{i \in M} \beta_i y_i - \sum_{i \in \{1, 2, \dots, L\}} \left( \gamma_i \sum_{j \in C'_i} y_j \right) - \delta \quad (6.23)$$

sujeto a:

$$\sum_{j \in V \setminus \{1\}} x_{1j} = 1 \quad (6.24)$$

$$\sum_{i \in V \setminus \{n\}} x_{in} = 1 \quad (6.25)$$

$$\sum_{j \in V \setminus \{n\}, i \neq j} x_{ji} = y_i \quad i \in V \setminus \{1\} \quad (6.26)$$

$$\sum_{j \in V \setminus \{1\}, i \neq j} x_{ij} = y_i \quad i \in V \setminus \{n\} \quad (6.27)$$

$$\sum_{i \in V \setminus \{n\}} \sum_{j \in V \setminus \{1\}} t_{ij} x_{ij} \leq t_{max} \quad (6.28)$$

$$z_{ij}^k \leq x_{ij} \quad i, j, k \in V, \quad i \neq n, j \neq 1, k \neq 1, n \quad (6.29)$$

$$\sum_{i \in V \setminus \{1\}} z_{1j}^k = 1 \quad k \in V \setminus \{1, n\} \quad (6.30)$$

$$\sum_{i \in V \setminus \{n\}} z_{in}^k = 1 - y_k \quad k \in V \setminus \{1, n\} \quad (6.31)$$

$$\sum_{i \in V \setminus \{k, n\}} z_{ik}^k = y_k \quad k \in V \setminus \{1, n\} \quad (6.32)$$

$$\sum_{i \in V \setminus \{1, k\}} z_{ki}^k = 0 \quad k \in V \setminus \{1, n\} \quad (6.33)$$

$$\sum_{i \in V \setminus \{n\}} z_{ij}^k - \sum_{i \in V \setminus \{1\}} z_{ji}^k = 0 \quad k \in V \setminus \{1, n\}, j \neq k \quad (6.34)$$

$$y_i \in \{0, 1\} \quad i \in V \quad (6.35)$$

$$x_{ij} \in \{0, 1\} \quad (i, j) \in E \quad (6.36)$$

$$z_{ij}^k \geq 0 \quad (i, j) \in E, k \in V \quad (6.37)$$

## 6.4 ALGORITMO DE GENERACIÓN DE COLUMNAS PARA EL OP CON RESTRICCCIONES ADICIONALES

El algoritmo de generación de columnas consiste en resolver el problema maestro reducido para obtener variables duales que alimentarán a la función objetivo del subproblema. Posteriormente, la solución obtenida por este último será una nueva columna que se añadirá al maestro. El proceso se repite iterativamente hasta que el valor óptimo de la función objetivo del subproblema sea menor o igual que cero o hasta cumplir un criterio de parada predeterminado.

### 6.4.1 INICIALIZACIÓN

Para resolver el problema maestro en la primera iteración del algoritmo, es necesario inicializar un cierto número de variables o columnas, lo cual puede hacerse mediante la generación de variables triviales o por medio de una heurística. En el procedimiento propuesto en esta tesis, la inicialización consiste en generar tantas columnas distintas como el número de restricciones del problema maestro por medio del algoritmo constructivo utilizado en el GRASP-VNS.

### 6.4.2 SOLUCIÓN DEL PROBLEMA MAESTRO

En cada iteración del método de generación de columnas, el problema maestro relajado ( $x_r$  continua) es resuelto a optimalidad haciendo uso del solver Cplex 12.6 y las variables duales así obtenidas son integradas en la función objetivo del subproblema.

### 6.4.3 SOLUCIÓN DEL SUBPROBLEMA

Para resolver el subproblema se utiliza una heurística constructiva para generar tantas columnas nuevas como el número de restricciones del problema maestro. Dicho

método hace uso de una función de evaluación para cada nodo  $i' \in V$ , dada por la ecuación (6.38).

$$e_{i'} = \frac{z_S(i')}{\min_{(i,j) \in r} \{t_{ii'} + t_{i'j} - t_{ij}\}} \quad (6.38)$$

Donde  $z_S(i')$  es igual al coeficiente del nodo  $i'$  en la función objetivo del subproblema y  $t_{ij}$  es el tiempo requerido para viajar del nodo  $i$  al nodo  $j$ , para todo  $(i, j) \in E$ .

Cabe mencionar que, debido a que se busca generar columnas con costo reducido positivo, solo se consideran como nodos candidatos a entrar en la solución a aquellos cuyo valor  $z_S(i')$  sea positivo.

El método utilizado para generar columnas de manera heurística es similar al utilizado en la fase constructiva del GRASP-VNS y está descrito por el Algoritmo 10.

---

**Algoritmo 10** Algoritmo constructivo para el subproblema

---

**Entrada:**

$I = (V, M, O, L, t, s, t_{max})$  : Instancia del problema

$z_S$  : Función objetivo del subproblema

1:  $LC := V$

2:  $r \leftarrow \{1, n\}$

3: **mientras**  $LC \neq \emptyset$  **hacer**

4:  $e_{i'} = \frac{z_S(i')}{\min_{(i,j) \in r} \{t_{ii'} + t_{i'j} - t_{ij}\}}, \forall i' \in CL$

5:  $e_{min} := \min\{e_{i'} | i' \in CL\}$

6:  $e_{max} := \max\{e_{i'} | i' \in CL\}$

7:  $LRC := \{i' \in CL | e_{i'} \geq (1 - \alpha)e_{max} + \alpha e_{min}\}$

8: Seleccionar  $k \in LRC$  aleatoriamente

9: **si**  $l(r) + e_k \leq t_{max}$ , **entonces**

10:     Insertar  $k$  en  $r$  en la posición correspondiente a la inserción más cercana

11:      $LC := LC \setminus \{k\}$

12: **fin si**

13: **fin mientras**

**Salida:**  $r$

---

Debido al uso de un algoritmo heurístico para generar múltiples columnas en

una iteración, es posible que se generen variables repetidas, por lo cual se aplica el algoritmo de filtrado de soluciones definido en la Sección 4.3.2 para evitar la repetición de variables en el problema maestro. Si el número de columnas nuevas es igual a cero, se resuelve el subproblema a optimalidad por medio del solver Cplex 12.6.

#### 6.4.4 CÁLCULO DE COTAS

Por medio de la ejecución del algoritmo de generación de columnas se hace posible la obtención de cotas tanto superior como inferior para el valor óptimo de la función objetivo problema maestro relajado.

Es evidente que en cierta iteración del algoritmo, una cota inferior para el problema maestro relajado  $\underline{z}_{RMP}$  está dada por la solución del problema maestro reducido relajado. Por otro lado, nótese que para la obtención de una cota superior  $\bar{z}_{RMP}$ , la función objetivo del problema maestro relajado no puede ser incrementada en más de  $z_S^*$  unidades.

Por lo cual, en cada iteración del algoritmo se cumple la siguiente condición:

$$\underline{z}_{MP} \leq z_{RMP}^* \leq \underline{z}_{RMP} + z_S^* \quad (6.39)$$

Sin embargo, debido a que en el algoritmo propuesto en esta tesis solo se conoce el valor óptimo de la función objetivo del subproblema cuando no se pueden generar nuevas columnas a través del método heurístico, no es posible conocer la cota superior para el valor óptimo de la función objetivo del problema maestro relajado en cada iteración. Con el propósito de aumentar el número de iteraciones en las cuales se tiene información acerca de la cota superior del problema maestro, se puede hacer la siguiente observación.

**Observación 6.1** *Si en una columna factible están contenidos todos los nodos cuyo*

*coeficiente en la función objetivo del subproblema es positivo, entonces la columna es una solución óptima del mismo.*

Por último, cabe destacar que si bien las cotas que se obtienen mediante el desarrollo del procedimiento corresponden al problema maestro relajado, la cota superior es, en efecto, una cota superior para el problema original; por otro lado, una cota inferior para el mismo puede ser encontrada al resolver el problema maestro reducido de la última iteración considerando a las variables  $x_k$  como binarias.

El algoritmo de generación de columnas propuesto en esta tesis se detiene cuando ha transcurrido una hora de ejecución o cuando el gap entre las cotas para el problema maestro relajado es menor que 5%.

## 6.5 RESULTADOS COMPUTACIONALES

El método de generación de columnas propuesto en esta tesis fue codificado en C++ y se realizó experimentación sobre los grafos cuyo número de nodos es menor que 50 en cada una de las seis clases de instancias en el mismo entorno experimental descrito en la Sección 3.2.

La Tabla 6.3 muestra el promedio por clase del gap entre las cotas superior e inferior para el problema original, del número de iteraciones y del tiempo de CPU requerido para obtener dichas cotas. Por último se presentan las medias de los mismos datos para el total de instancias en las cuales se realizó la experimentación.

Clase	Gap entre cotas	Número de iteraciones	Tiempo de CPU (segundos)
1	8.04	34.11	2685.99
2	3.50	79.63	1711.70
3	10.56	99.58	5182.42
4	11.78	19.61	4071.86
5	20.29	44.33	6270.83
6	12.99	46.38	6492.51
Total	11.12	52.51	4348.68

Tabla 6.3: Resultados obtenidos por el método de generación de columnas

De dicha tabla se puede observar que para las instancias de la clase 2 el algoritmo proporciona cotas de mejor calidad que en el resto de las instancias.

Para corroborar este hecho, se realizó una prueba no paramétrica de Kruskal Wallis, en la cual cada clase es vista como un tratamiento distinto y el gap entre las cotas de cada instancia es visto como una observación. Se probó la hipótesis nula de que la mediana del gap entre las cotas en cada clase es igual, contra la alternativa de que, al menos, una de las medianas es diferente. Dicha hipótesis está expresada por las ecuaciones (6.40) y (6.41).

$$H_0 : Me_1 = Me_2 = Me_3 = Me_4 = Me_5 = Me_6 \quad (6.40)$$

$$H_1 : Me_i \neq Me_j, \text{ para } i \neq j, i \quad (6.41)$$

De la Tabla 6.4 se puede concluir que, al observar un valor  $p$  despreciable, se rechaza la hipótesis nula de que no existen diferencias significativas entre las medianas del gap entre cotas para las diferentes clases contra la alternativa de que al menos una mediana es distinta. Por lo tanto, en al menos una clase, el método de generación de columnas proporciona cotas más ajustadas que en el resto de las

Clase	N	Mediana	Clasificación del promedio	Z
1	28	4.771390628	77.3	0.10
2	24	0.671107833	52.8	-2.88
3	24	0.000000000	56.6	-2.41
4	28	3.665088479	79.0	0.33
5	24	1.47074E+01	102.6	3.17
6	24	6.207389303	90.3	1.67
General	152		76.5	

$H = 23.48$     $GL = 5$     $P = 0.000$

Tabla 6.4: Prueba de Kruskal Wallis: Gap entre cotas contra clase de instancias

clases.

Con el fin de determinar en cuál o cuáles instancias de las clases se obtienen cotas de mejor calidad, se obtuvieron los intervalos de confianza para las medias del gap entre cotas para cada clase, utilizando el ajuste de Bonferroni, como se muestra en la Tabla 6.5 y en la Figura 6.1.

Clase	Límite inferior	Límite superior
1	3.06479	13.02223
2	0.215411	6.7856
3	-2.12516	23.2371
4	3.21912	20.3358
5	8.6691	31.9083
6	4.89478	21.0801

Tabla 6.5: Intervalos de confianza para la media del gap entre cotas en cada clase con una confiabilidad del 95 %

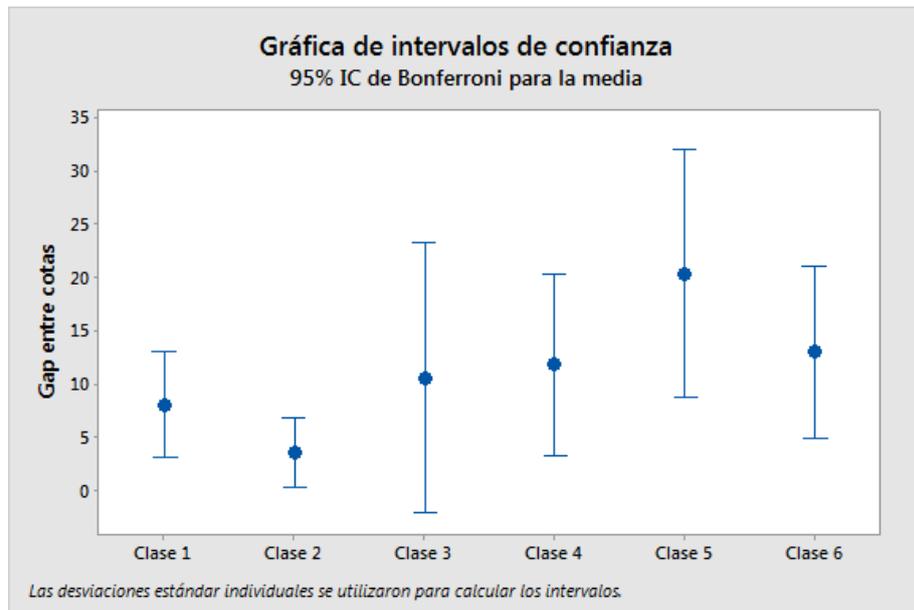


Figura 6.1: Intervalos de confianza para el gap entre cotas en cada clase con una confiabilidad del 95 %

De la obtención de dichos intervalos de confianza se concluye que, en efecto, el gap entre las cotas obtenidas para las instancias de la clase 2 es significativamente menor que el de las de la clase 5. Sin embargo, no se pueden establecer conclusiones ente el resto de las clases.

### 6.5.1 COMPARACIÓN DE LAS COTAS OBTENIDAS CONTRA EL VALOR ÓPTIMO

En la sección anterior se presentaron resultados que muestran que el promedio del gap entre las cotas superior e inferior obtenidas en cada clase por el procedimiento de generación de columnas puede llegar a superar el 30% (véase Tabla 6.5). Sin embargo, estos resultados no brindan información acerca de la calidad de las cotas superiores e inferiores por sí solas.

En la Tabla 6.6 se muestran los promedios del gap de las cotas obtenidas con respecto al óptimo en cada clase y en la totalidad de las instancias sobre las cuales se ejecutó el algoritmo.

Clase	Cota inferior	Cota superior
1	3.33	5.02
2	0.00	3.50
3	0.00	10.56
4	3.28	8.82
5	1.69	18.99
6	2.51	10.95
Total	1.88	9.50

Tabla 6.6: Gap de las cotas con respecto al óptimo

De la tabla se puede concluir que, en apariencia, la calidad del gap entre las cotas se ve afectada por la baja calidad de la cota superior, lo cual es un resultado esperado por el hecho de que el número de iteraciones del método de generación de columnas en las cuales se tiene información acerca de dicha cota es reducido.

Mas aún, es posible que haya iteraciones del algoritmo en las cuales la heurística encuentra la columna óptima y el método no lo detecta. Por lo cual una estrategia futura para mejorar la calidad de la cota superior es encontrar más reglas similares a la Observación 6.1 para detectar cuando una solución proporcionada por la heurística es el óptimo del subproblema.

Otra propuesta es la de desarrollar una heurística dual, lo cual proporcionaría cotas superiores para la función objetivo del subproblema y, en consecuencia, en cada iteración del algoritmo de generación de columnas se podrían obtener cotas superiores válidas.

Por último, otra línea de trabajo consiste en resolver el subproblema por medio de una heurística y por medio de un método exacto distinto al *branch and bound* de Cplex en cada iteración del algoritmo.

## CONCLUSIONES Y TRABAJO FUTURO

---

En este capítulo se presentan las conclusiones del trabajo desarrollado y se extienden líneas de investigación futura que permitan mejorar la calidad de los resultados obtenidos y que amplíen el alcance de la investigación.

### 7.1 CONCLUSIONES

El *Orienteering Problem* es un problema NP-duro, por lo cual el problema que se estudió en esta tesis, al ser un caso general del mismo, también lo es. Lo anterior implica que sea relevante el desarrollo de algoritmos que permitan la resolución del problema de manera eficiente.

En la literatura existen múltiples propuestas de resolución del OP y de variantes del mismo, tanto heurísticas como exactas; asimismo, es posible encontrar aplicaciones de problemas afines a problemáticas reales. No obstante, la versión del OP que fue estudiada en esta tesis no ha sido previamente estudiada.

En este trabajo se presentaron dos formulaciones matemáticas para el OP con restricciones adicionales que difieren en la manera en que es abordada la eliminación de ciclos. Una contiene restricciones basadas en flujo de múltiples productos y la otra, restricciones secuenciales. La formulación basada en flujo posee un mayor número de restricciones y de variables; sin embargo, fue capaz de resolver a optimalidad una mayor cantidad de instancias que la segunda y se mostró de manera experimental

que su relajación lineal provee cotas superiores más ajustadas.

Por otro lado, se propuso un algoritmo metaheurístico híbrido basado en un esquema GRASP reactivo, cuya fase de postprocesamiento de soluciones se basa en una metodología VNS general. Dicho híbrido alcanzó soluciones óptimas o cercanas al óptimo en los casos para los cuales se pudo resolver el problema haciendo uso del solver Cplex 12.6.

Adicional al método heurístico, se propuso un algoritmo de generación de columnas. Aunque dicho algoritmo no provee buenas cotas superiores al problema, al presentar un gap promedio con respecto al óptimo de 9.5%; la calidad de las cotas inferiores que proporcionó es buena, obteniendo un gap promedio con respecto al óptimo de 1.88%.

## 7.2 TRABAJO FUTURO

Como trabajo a futuro se propone modificar el algoritmo de generación de columnas propuesto en el capítulo 6 con el propósito de mejorar la calidad de la cota superior del problema maestro. En particular, para ello se pretende abordar de manera diferente la solución del subproblema.

Una idea es desarrollar una heurística para el dual del subproblema, lo cual proporcionaría cotas superiores para el mayor costo reducido y, en consecuencia, cotas superiores válidas para el problema maestro en cada iteración del algoritmo de generación de columnas. Dicho algoritmo pretende combinarse con la solución del subproblema a optimalidad, en el caso de que se requiera, pero utilizando un algoritmo de optimización más afín al problema que el *branch and bound* de Cplex, como lo puede ser programación dinámica.

Por otro lado, se pretende realizar la extensión de los algoritmos propuestos para obtener buenas soluciones para el *Team Orienteering Problem* con restricciones adicionales. En el caso del método heurístico, se planea desarrollar nuevas vecindades

que comprendan movimientos entre rutas y en el del algoritmo exacto, basta con agregar restricciones en el problema maestro. Además, se buscará integrar el esquema de generación de columnas en un método tipo *branch and price*.

Por último, se desea aplicar los algoritmos propuestos a la resolución de un caso de estudio que comprende el soporte en la planificación de las rutas de servicio en una empresa dedicada a la recolección y transporte de residuos peligrosos. Donde cada cliente tiene una puntuación asociada a la urgencia con la que debe recibir el servicio de recolección de residuos y la duración del recorrido no debe exceder la duración de la jornada laboral de los empleados. Adicional a lo anterior, existen servicios que no pueden ser postergados para otro día, debido al peligro que representa para los clientes de la empresa el mantener los residuos. Por otro lado, existen residuos que son incompatibles con otros y que no pueden ser transportados en la misma unidad.

Cabe destacar que en este caso de estudio, es posible que los clientes establezcan horarios en los cuales deben ser atendidos, por lo cual, debe tomarse en cuenta la existencia de ventanas de tiempo.

## APÉNDICE A

# TABLAS DE RESULTADOS

---

En este apéndice se presentan las tablas que contienen los resultados de los experimentos computacionales realizados en este trabajo.

## A.1 TABLAS REFERENTES A PRUEBAS ENTRE LOS MODELOS

Las Tablas A.1 a A.6 muestran los resultados obtenidos por Cplex 12.6 al ejecutar los modelos M1 y M2 con un tiempo límite de una hora en las clases 1 a 6, respectivamente. Se muestra el valor de la función objetivo alcanzado en cada modelo y se indica si es óptima o si es una solución entera factible no óptima, así como el tiempo requerido para encontrar dicha solución.

En los casos en los que no se muestra ningún valor es porque el solver no encontró ninguna solución entera al alcanzar el límite de tiempo y, en las instancias en las cuales se encontró una solución entera, se muestra el gap reportado por Cplex.

Instancia	M1				M2			
	Objetivo	Estatus	Gap	Tiempo (segs.)	Objetivo	Estatus	Gap	Tiempo (segs.)
att48A	15	Óptima	0	1570.424	15	Óptima	0	73.958
att48B	17	Óptima	0	11.178	-	-	-	>3600
att48C	8	Entera	0.3333	>3600	9	Óptima	0	254.551
att48D	13	Óptima	0	1314.599	13	Óptima	0	43.42
att48E	14	Óptima	0	2659.148	14	Óptima	0	66.16
cmt121A	-	-	-	>3600	-	-	-	>3600
cmt121B	-	-	-	>3600	-	-	-	>3600
cmt121C	-	-	-	>3600	-	-	-	>3600
cmt121D	-	-	-	>3600	-	-	-	>3600
cmt151A	-	-	-	>3600	-	-	-	>3600
cmt151B	872	Óptima	0	889.38	-	-	-	>3600
cmt151C	-	-	-	>3600	-	-	-	>3600
cmt151D	-	-	-	>3600	-	-	-	>3600
cmt151E	-	-	-	>3600	-	-	-	>3600
cmt200A	-	-	-	>3600	-	-	-	>3600
cmt200B	-	-	-	>3600	-	-	-	>3600
cmt200B	-	-	-	>3600	-	-	-	>3600
cmt200C	-	-	-	>3600	-	-	-	>3600
cmt200D	-	-	-	>3600	-	-	-	>3600
eil30A	6375	Óptima	0	2.137	6375	Óptima	0	8.761
eil30B	5125	Óptima	0	2601.934	5125	Óptima	0	50.74
eil30C	5775	Óptima	0	1135.014	5775	Óptima	0	19.372
eil30D	6275	Óptima	0	53.332	6275	Óptima	0	10.472
eil33A	5230	Entera	0.5609	>3600	5230	Óptima	0	120.505
eil33B	14380	Óptima	0	41.303	14380	Óptima	0	7.757
eil33C	6880	Entera	0.4531	>3600	7430	Óptima	0	99.073
eil33D	11630	Entera	0.0935	>3600	11630	Óptima	0	19.936
eil33E	12830	Entera	0.0415	>3600	12830	Óptima	0	29.883
eil51A	245	Óptima	0	1009.554	245	Óptima	0	108.898
eil51B	287	Óptima	0	6.408	287	Óptima	0	308.901
eil51C	122	Entera	0.2173	>3600	122	Óptima	0	566.98
eil51D	150	Entera	0.2367	>3600	150	Óptima	0	118.979
eil51E	177	Óptima	0	516.879	177	Óptima	0	351.381
eil76A	507	Entera	0.0482	>3600	520	Óptima	0	2206.362
eil76B	599	Óptima	0	24.664	-	-	-	>3600
eil76C	-	-	-	>3600	232	Óptima	0	2801.548
eil76D	-	-	-	>3600	312	Óptima	0	2608.895
eil76E	-	-	-	>3600	364	Entera	0.0123	>3600
eil101A	-	-	-	>3600	-	-	-	>3600
eil101B	612	Óptima	0	244.971	-	-	-	>3600

La tabla continúa en la página siguiente

Instancia	M1				M2			
	Objetivo	Estatus	Gap	Tiempo (segs.)	Objetivo	Estatus	Gap	Tiempo (segs.)
eil101C	-	-	-	>3600	-	-	-	>3600
eil101D	-	-	-	>3600	-	-	-	>3600
eil101E	-	-	-	>3600	-	-	-	>3600
gil262A	-	-	-	>3600	-	-	-	>3600
gil262B	-	-	-	>3600	-	-	-	>3600
gil262C	-	-	-	>3600	-	-	-	>3600
gil262D	-	-	-	>3600	-	-	-	>3600
gil262E	-	-	-	>3600	-	-	-	>3600
op21A	165	Óptima	0	0.349	165	Óptima	0	2.253
op21B	135	Óptima	0	0.351	135	Óptima	0	1.133
op21C	150	Óptima	0	0.655	150	Óptima	0	0.82
op21D	155	Óptima	0	0.748	155	Óptima	0	1.834
op32A	85	Óptima	0	14.007	85	Óptima	0	5.691
op32B	115	Óptima	0	0.181	115	Óptima	0	13.999
op32C	35	Óptima	0	95.094	35	Óptima	0	3.741
op32D	60	Óptima	0	20.716	60	Óptima	0	4.243
op32E	75	Óptima	0	20.359	75	Óptima	0	5.734
op33A	260	Óptima	0	3.869	260	Óptima	0	3.583
op33B	330	Óptima	0	1.218	330	Óptima	0	3.735
op33C	110	Óptima	0	1.622	110	Óptima	0	4.008
op33D	160	Óptima	0	5.054	160	Óptima	0	4.657
op33E	180	Óptima	0	0.614	180	Óptima	0	3.935

Tabla A.1: Resultados obtenidos por Cplex 12.6 haciendo uso de los modelos propuestos en las instancias de la clase 1

Instancia	M1				M2			
	Objetivo	Estatus	Gap	Tiempo (segs.)	Objetivo	Estatus	Gap	Tiempo (segs.)
att48A	12	Óptima	0	20.214	12	Óptima	0	54.688
att48B	11	Entera	0.0833	>3600	11	Óptima	0	43.836
att48C	12	Óptima	0	36.654	12	Óptima	0	520.818
att48D	12	Óptima	0	16.284	12	Óptima	0	127.328
cmt121A	-	-	-	>3600	-	-	-	>3600
cmt121B	-	-	-	>3600	-	-	-	>3600
cmt121C	-	-	-	>3600	-	-	-	>3600
cmt121D	-	-	-	>3600	-	-	-	>3600
cmt151A	-	-	-	>3600	-	-	-	>3600
cmt151B	-	-	-	>3600	-	-	-	>3600
cmt151C	-	-	-	>3600	-	-	-	>3600
cmt151D	-	-	-	>3600	-	-	-	>3600

La tabla continúa en la página siguiente

Instancia	M1				M2			
	Objetivo	Estatus	Gap	Tiempo (segs.)	Objetivo	Estatus	Gap	Tiempo (segs.)
cmt151E	-	-	-	>3600	-	-	-	>3600
cmt200A	-	-	-	>3600	-	-	-	>3600
cmt200B	-	-	-	>3600	-	-	-	>3600
cmt200C	-	-	-	>3600	-	-	-	>3600
cmt200D	-	-	-	>3600	-	-	-	>3600
eil30A	2925	Óptima	0	47.788	2925	Óptima	0	3.847
eil30B	-	-	-	>3600	1625	Óptima	0	3.599
eil30C	1625	Óptima	0	164.868	1625	Óptima	0	5.92
eil30D	2475	Óptima	0	70.43	2475	Óptima	0	5.727
eil33A	8480	Óptima	0	513.4	8480	Óptima	0	4.985
eil33B	8480	Óptima	0	964.152	8480	Óptima	0	2.52
eil33C	8480	Óptima	0	1294.747	8480	Óptima	0	13.946
eil33D	9430	Óptima	0	3.332	9430	Óptima	0	6.112
eil51A	168	Óptima	0	2.824	168	Óptima	0	63.403
eil51B	148	Entera	0.119	>3600	148	Óptima	0	134.482
eil51C	168	Óptima	0	17.72	168	Óptima	0	27.935
eil51D	168	Óptima	0	2.26	168	Óptima	0	85.405
eil76A	-	-	-	>3600	354	Entera	0.0816	>3600
eil76B	409	Óptima	0	5.332	409	Óptima	0	3084.084
eil76C	-	-	-	>3600	201	Óptima	0	593.724
eil76D	293	Entera	0.0844	>3600	294	Óptima	0	798.089
eil76E	-	-	-	>3600	345	Óptima	0	1135.908
eil101A	-	-	-	>3600	-	-	-	>3600
eil101B	409	Óptima	0	138.441	-	-	-	>3600
eil101C	-	-	-	>3600	-	-	-	>3600
eil101D	-	-	-	>3600	-	-	-	>3600
eil101E	-	-	-	>3600	-	-	-	>3600
gil262A	-	-	-	>3600	-	-	-	>3600
gil262B	-	-	-	>3600	-	-	-	>3600
gil262C	-	-	-	>3600	-	-	-	>3600
gil262D	-	-	-	>3600	-	-	-	>3600
op21A	60	Óptima	0	0.209	60	Óptima	0	0.527
op21B	125	Óptima	0	0.128	125	Óptima	0	0.588
op21C	60	Óptima	0	0.221	60	Óptima	0	0.766
op21D	75	Óptima	0	0.305	75	Óptima	0	0.996
op21E	80	Óptima	0	0.413	80	Óptima	0	1.523
op32A	80	Óptima	0	0.565	80	Óptima	0	4.392
op32B	70	Óptima	0	1.292	70	Óptima	0	2.27
op32C	80	Óptima	0	1.879	80	Óptima	0	4.533
op33A	210	Óptima	0	2.822	210	Óptima	0	1.169

La tabla continúa en la página siguiente

Instancia	M1				M2			
	Objetivo	Estatus	Gap	Tiempo (segs.)	Objetivo	Estatus	Gap	Tiempo (segs.)
op33B	170	Óptima	0	12.212	170	Óptima	0	1.539
op33C	210	Óptima	0	1.029	210	Óptima	0	1.701
op33D	220	Óptima	0	0.231	220	Óptima	0	3.695

Tabla A.2: Resultados obtenidos por Cplex 12.6 haciendo uso de los modelos propuestos en las instancias de la clase 2

Instancia	M1				M2			
	Objetivo	Estatus	Gap	Tiempo (segs.)	Objetivo	Estatus	Gap	Tiempo (segs.)
att48A	4	Entera	0.2	>3600	4	Óptima	0	37.167
att48B	7	Óptima	0	3.497	7	Óptima	0	125.375
att48C	4	Entera	0.2	>3600	4	Óptima	0	40.052
att48D	5	Óptima	0	37.015	5	Óptima	0	31.17
att48E	5	Óptima	0	2669.351	5	Óptima	0	19.806
cmt121A	-	-	-	>3600	-	-	-	>3600
cmt121B	-	-	-	>3600	-	-	-	>3600
cmt121C	-	-	-	>3600	-	-	-	>3600
cmt121D	-	-	-	>3600	-	-	-	>3600
cmt151A	-	-	-	>3600	-	-	-	>3600
cmt151B	-	-	-	>3600	-	-	-	>3600
cmt151C	-	-	-	>3600	-	-	-	>3600
cmt151D	-	-	-	>3600	-	-	-	>3600
cmt200A	-	-	-	>3600	-	-	-	>3600
cmt200B	-	-	-	>3600	-	-	-	>3600
cmt200C	-	-	-	>3600	-	-	-	>3600
cmt200D	-	-	-	>3600	-	-	-	>3600
eil30A	950	Óptima	0	403.515	950	Óptima	0	2.012
eil30B	1950	Óptima	0	73.75	1950	Óptima	0	2.025
eil30C	2075	Óptima	0	0.293	2075	Óptima	0	1.083
eil30D	2075	Óptima	0	0.125	2075	Óptima	0	1.028
eil33A	5530	Óptima	0	113.885	5530	Óptima	0	1.172
eil33B	5530	Óptima	0	3.014	5530	Óptima	0	1.416
eil33C	5530	Óptima	0	0.086	5530	Óptima	0	2.06
eil51A	99	Óptima	0	1.811	99	Óptima	0	29.912
eil51B	85	Entera	0.1169	>3600	90	Óptima	0	77.196
eil51C	99	Óptima	0	6.757	99	Óptima	0	20.814
eil51D	99	Óptima	0	3.902	99	Óptima	0	25.281
eil76A	151	Entera	0.1592	>3600	167	Óptima	0	37.087
eil76B	202	Óptima	0	5.833	202	Óptima	0	936.951
eil76C	-	-	-	>3600	142	Óptima	0	142.45

La tabla continúa en la página siguiente

Instancia	M1				M2			
	Objetivo	Estatus	Gap	Tiempo (segs.)	Objetivo	Estatus	Gap	Tiempo (segs.)
eil76D	-	-	-	>3600	183	Entera	0.0941	>3600
eil76E	202	Óptima	0	34.737	202	Óptima	0	2775.254
eil101A	175	Óptima	0	466.002	-	-	-	>3600
eil101B	-	-	-	>3600	-	-	-	>3600
eil101C	-	-	-	>3600	-	-	-	>3600
eil101D	175	Óptima	0	1175.793	-	-	-	>3600
gil262A	-	-	-	>3600	-	-	-	>3600
gil262B	-	-	-	>3600	-	-	-	>3600
gil262C	-	-	-	>3600	-	-	-	>3600
gil262D	-	-	-	>3600	-	-	-	>3600
op21A	45	Óptima	0	0.029	45	Óptima	0	0.108
op21B	30	Óptima	0	0.064	30	Óptima	0	0.167
op21C	45	Óptima	0	0.042	45	Óptima	0	0.116
op21D	45	Óptima	0	0.031	45	Óptima	0	0.103
op32A	45	Óptima	0	0.458	45	Óptima	0	2.599
op32B	45	Óptima	0	0.325	45	Óptima	0	1.671
op32C	45	Óptima	0	0.184	45	Óptima	0	1.8
op32D	45	Óptima	0	0.438	45	Óptima	0	1.566
op33A	130	Óptima	0	0.063	130	Óptima	0	1.116
op33B	80	Óptima	0	0.4	80	Óptima	0	0.95
op33C	120	Óptima	0	1.056	120	Óptima	0	1.056
op33D	130	Óptima	0	0.111	130	Óptima	0	0.878

Tabla A.3: Resultados obtenidos por Cplex 12.6 haciendo uso de los modelos propuestos en las instancias de la clase 3

Instancia	M1				M2			
	Objetivo	Estatus	Gap	Tiempo (segs.)	Objetivo	Estatus	Gap	Tiempo (segs.)
att48A	20	Entera	0.1556	>3600	21	Óptima	0	133.127
att48B	27	Entera	0.129	>3600	29	Óptima	0	474.224
att48C	10	Entera	0.4444	>3600	13	Óptima	0	250.484
att48D	16	Entera	0.2381	>3600	17	Óptima	0	145.796
att48E	19	Entera	0.1364	>3600	20	Óptima	0	108.115
cmt121A	-	-	-	>3600	-	-	-	>3600
cmt121B	-	-	-	>3600	-	-	-	>3600
cmt121C	-	-	-	>3600	-	-	-	>3600
cmt121D	-	-	-	>3600	-	-	-	>3600
cmt151A	-	-	-	>3600	-	-	-	>3600
cmt151B	-	-	-	>3600	-	-	-	>3600
cmt151C	-	-	-	>3600	-	-	-	>3600

La tabla continúa en la página siguiente

Instancia	M1				M2			
	Objetivo	Estatus	Gap	Tiempo (segs.)	Objetivo	Estatus	Gap	Tiempo (segs.)
cmt151D	-	-	-	>3600	-	-	-	>3600
cmt151E	-	-	-	>3600	-	-	-	>3600
cmt200A	-	-	-	>3600	-	-	-	>3600
cmt200B	-	-	-	>3600	-	-	-	>3600
cmt200C	-	-	-	>3600	-	-	-	>3600
cmt200D	-	-	-	>3600	-	-	-	>3600
cmt200E	-	-	-	>3600	-	-	-	>3600
eil30A	9350	Entera	0.0469	>3600	9375	Óptima	0	125.55
eil30B	5350	Entera	0.408	>3600	5450	Óptima	0	163.531
eil30C	7300	Entera	0.215	>3600	7300	Óptima	0	23.758
eil30D	8575	Entera	0.104	>3600	8575	Óptima	0	33.191
eil33A	6250	Entera	0.637	>3600	6950	Óptima	0	120.358
eil33B	19730	Entera	0.0775	>3600	19730	Óptima	0	79.453
eil33C	10630	Entera	0.4096	>3600	10630	Óptima	0	25.407
eil33D	13830	Entera	0.2733	>3600	14180	Óptima	0	51.876
eil33E	17430	Entera	0.1235	>3600	17430	Óptima	0	15.225
eil51A	328	Entera	0.0505	>3600	328	Óptima	0	755.242
eil51B	509	Óptima	0	1596.337	509	Óptima	0	611.346
eil51C	-	-	-	>3600	168	Óptima	0	751.518
eil51D	201	Óptima	0	420.812	201	Óptima	0	467.732
eil51E	232	Óptima	0	868.368	232	Óptima	0	1950.282
eil76A	-	-	-	>3600	514	Entera	0.1937	>3600
eil76B	910	Entera	0.0269	>3600	923	Óptima	0	3314.444
eil76C	-	-	-	>3600	269	Entera	0.0389	>3600
eil76D	311	Entera	0.285	>3600	-	-	-	>3600
eil76E	-	-	-	>3600	-	-	-	>3600
eil101A	730	Entera	0.1141	>3600	-	-	-	>3600
eil101B	-	-	-	>3600	-	-	-	>3600
eil101C	-	-	-	>3600	-	-	-	>3600
eil101D	-	-	-	>3600	-	-	-	>3600
eil101E	-	-	-	>3600	-	-	-	>3600
gil262A	-	-	-	>3600	-	-	-	>3600
gil262B	-	-	-	>3600	-	-	-	>3600
gil262C	-	-	-	>3600	-	-	-	>3600
gil262D	-	-	-	>3600	-	-	-	>3600
gil262E	-	-	-	>3600	-	-	-	>3600
op21A	260	Óptima	0	12.427	260	Óptima	0	5.948
op21B	180	Óptima	0	4.285	180	Óptima	0	1.314
op21C	215	Óptima	0	5.581	215	Óptima	0	1.878
op21D	260	Óptima	0	4.185	260	Óptima	0	1.5

La tabla continúa en la página siguiente

Instancia	M1				M2			
	Objetivo	Estatus	Gap	Tiempo (segs.)	Objetivo	Estatus	Gap	Tiempo (segs.)
op32A	110	Óptima	0	23.845	110	Óptima	0	6.785
op32B	180	Óptima	0	7.925	180	Óptima	0	4.61
op32C	35	Óptima	0	1553.496	35	Óptima	0	3.285
op32D	75	Óptima	0	44.213	75	Óptima	0	2.701
op32E	90	Óptima	0	67.337	90	Óptima	0	3.263
op33A	310	Óptima	0	6.059	310	Óptima	0	8.841
op33B	460	Óptima	0	5.689	460	Óptima	0	8.576
op33C	120	Óptima	0	33.667	120	Óptima	0	5.782
op33D	180	Óptima	0	12.654	180	Óptima	0	6.73
op33E	220	Óptima	0	4.325	220	Óptima	0	7.468

Tabla A.4: Resultados obtenidos por Cplex 12.6 haciendo uso de los modelos propuestos en las instancias de la clase 4

Instancia	M1				M2			
	Objetivo	Estatus	Gap	Tiempo (segs.)	Objetivo	Estatus	Gap	Tiempo (segs.)
att48A	-	-	-	>3600	24	Óptima	-	184.495
att48B	-	-	-	>3600	19	Óptima	-	414.642
att48C	-	-	-	>3600	23	Óptima	-	184.723
att48D	22	Entera	0.1852	>3600	25	Óptima	-	174.804
cmt121A	-	-	-	>3600	-	-	-	>3600
cmt121B	-	-	-	>3600	-	-	-	>3600
cmt121C	-	-	-	>3600	-	-	-	>3600
cmt121D	-	-	-	>3600	-	-	-	>3600
cmt151A	-	-	-	>3600	-	-	-	>3600
cmt151B	-	-	-	>3600	-	-	-	>3600
cmt151C	-	-	-	>3600	-	-	-	>3600
cmt151D	-	-	-	>3600	-	-	-	>3600
cmt151E	-	-	-	>3600	-	-	-	>3600
cmt200A	-	-	-	>3600	-	-	-	>3600
cmt200B	-	-	-	>3600	-	-	-	>3600
cmt200C	-	-	-	>3600	-	-	-	>3600
cmt200D	-	-	-	>3600	-	-	-	>3600
eil30A	5075	Entera	0.3095	>3600	6225	Óptima	0	59.378
eil30B	3350	Entera	0.4675	>3600	3350	Óptima	0	19.091
eil30C	4750	Entera	0.2717	>3600	4750	Óptima	0	19.724
eil30D	5075	Entera	0.2698	>3600	5075	Óptima	0	93.202
eil33A	14230	Entera	0.1443	>3600	14230	Óptima	0	47.853
eil33B	-	-	-	>3600	12880	Óptima	0	65.6
eil33C	14930	Entera	0.1155	>3600	14930	Óptima	0	7.877

La tabla continúa en la página siguiente

Instancia	M1				M2			
	Objetivo	Estatus	Gap	Tiempo (segs.)	Objetivo	Estatus	Gap	Tiempo (segs.)
eil33D	-	-	-	>3600	15680	Óptima	0	82.882
eil51A	402	Óptima	0	277.586	402	Óptima	0	424.854
eil51B	-	-	-	>3600	236	Óptima	0	1027.565
eil51C	-	-	-	>3600	302	Óptima	0	1252.081
eil51D	-	-	-	>3600	357	Óptima	0	840.48
eil76A	-	-	-	>3600	-	-	-	>3600
eil76B	747	Entera	0.0132	>3600	-	-	-	>3600
eil76C	-	-	-	>3600	-	-	-	>3600
eil76D	-	-	-	>3600	-	-	-	>3600
eil76E	-	-	-	>3600	-	-	-	>3600
eil101A	529	Entera	0.1045	>3600	-	-	-	>3600
eil101B	855	Entera	0.0089	>3600	-	-	-	>3600
eil101C	-	-	-	>3600	-	-	-	>3600
eil101D	-	-	-	>3600	-	-	-	>3600
eil101E	-	-	-	>3600	-	-	-	>3600
gil262A	-	-	-	>3600	-	-	-	>3600
gil262B	-	-	-	>3600	-	-	-	>3600
gil262C	-	-	-	>3600	-	-	-	>3600
gil262D	-	-	-	>3600	-	-	-	>3600
op21A	135	Óptima	0	0.239	135	Óptima	0	1.078
op21B	235	Óptima	0	12.013	235	Óptima	0	2.52
op21C	135	Óptima	0	0.278	135	Óptima	0	1.158
op21D	135	Óptima	0	0.446	135	Óptima	0	4.68
op21E	165	Óptima	0	2.84	165	Óptima	0	3.413
op32A	150	Óptima	0	7.202	150	Óptima	0	12.966
op32B	110	Óptima	0	8.077	110	Óptima	0	15.112
op32C	140	Óptima	0	18.905	140	Óptima	0	10.905
op33A	400	Óptima	0	0.954	400	Óptima	0	28.387
op33B	270	Óptima	0	2388.509	270	Óptima	0	26.176
op33C	410	Óptima	0	35.717	410	Óptima	0	20.342
op33D	440	Óptima	0	6.246	440	Óptima	0	15.243

Tabla A.5: Resultados obtenidos por Cplex 12.6 haciendo uso de los modelos propuestos en las instancias de la clase 5

Instancia	M1				M2			
	Objetivo	Estatus	Gap	Tiempo (segs.)	Objetivo	Estatus	Gap	Tiempo (segs.)
att48A	-	-	-	>3600	11	Óptima	0	225.673
att48B	17	Entera	0.1905	>3600	19	Óptima	0	735.092
att48C	10	Entera	0.2857	>3600	12	Óptima	0	181.001

La tabla continúa en la página siguiente

Instancia	M1				M2			
	Objetivo	Estatus	Gap	Tiempo (segs.)	Objetivo	Estatus	Gap	Tiempo (segs.)
att48D	14	Entera	0.1244	>3600	14	Óptima	0	495.577
att48E	14	Entera	0.1765	>3600	16	Óptima	0	284.135
cmt121A	-	-	-	>3600	-	-	-	>3600
cmt121B	-	-	-	>3600	-	-	-	>3600
cmt121C	-	-	-	>3600	-	-	-	>3600
cmt121D	-	-	-	>3600	-	-	-	>3600
cmt151A	-	-	-	>3600	-	-	-	>3600
cmt151B	-	-	-	>3600	-	-	-	>3600
cmt151C	-	-	-	>3600	-	-	-	>3600
cmt151D	-	-	-	>3600	-	-	-	>3600
cmt200A	-	-	-	>3600	-	-	-	>3600
cmt200B	-	-	-	>3600	-	-	-	>3600
cmt200C	-	-	-	>3600	-	-	-	>3600
cmt200D	-	-	-	>3600	-	-	-	>3600
eil30A	-	-	-	>3600	1550	Óptima	0	31.864
eil30B	3900	Entera	0.3185	>3600	4775	Óptima	0	28.534
eil30C	5275	Entera	0.0983	>3600	5550	Óptima	0	15.69
eil30D	5850	Óptima	0	2639.199	5850	Óptima	0	18.232
eil33A	-	-	-	>3600	10600	Óptima	0	32.379
eil33B	-	-	-	>3600	13330	Óptima	0	61.308
eil33C	14030	Óptima	0	3313.208	14030	Óptima	0	27.287
eil51A	329	Óptima	0	175.542	329	Óptima	0	511.914
eil51B	-	-	-	>3600	212	Óptima	0	136.532
eil51C	-	-	-	>3600	289	Óptima	0	221.036
eil51D	-	-	-	>3600	344	Óptima	0	267.911
eil76A	-	-	-	>3600	228	Entera	0.0727	>3600
eil76B	-	-	-	>3600	-	-	-	>3600
eil76C	-	-	-	>3600	211	Óptima	0	3134.232
eil76D	-	-	-	>3600	-	-	-	>3600
eil76E	-	-	-	>3600	-	-	-	>3600
eil101A	631	Entera	0.0107	>3600	-	-	-	>3600
eil101B	-	-	-	>3600	-	-	-	>3600
eil101C	-	-	-	>3600	-	-	-	>3600
eil101D	-	-	-	>3600	-	-	-	>3600
gil262A	-	-	-	>3600	-	-	-	>3600
gil262B	-	-	-	>3600	-	-	-	>3600
gil262C	-	-	-	>3600	-	-	-	>3600
gil262D	-	-	-	>3600	-	-	-	>3600
op21A	155	Óptima	0	0.91	155	Óptima	0	2.993
op21B	75	Óptima	0	0.345	75	Óptima	0	1.02

La tabla continúa en la página siguiente

Instancia	M1				M2			
	Objetivo	Estatus	Gap	Tiempo (segs.)	Objetivo	Estatus	Gap	Tiempo (segs.)
op21C	110	Óptima	0	0.538	110	Óptima	0	1.73
op21D	160	Óptima	0	1.073	160	Óptima	0	5.951
op32A	115	Óptima	0	1.24	115	Óptima	0	4.425
op32B	65	Óptima	0	1321.154	65	Óptima	0	4.3
op32C	105	Óptima	0	2156.601	105	Óptima	0	3.861
op32D	130	Óptima	0	6.962	130	Óptima	0	5.158
op33A	350	Óptima	0	0.888	350	Óptima	0	2.74
op33B	130	Óptima	0	5.682	130	Óptima	0	18.044
op33C	240	Óptima	0	21.416	240	Óptima	0	5.557
op33D	300	Óptima	0	5.863	300	Óptima	0	3.175

Tabla A.6: Resultados obtenidos por Cplex 12.6 haciendo uso de los modelos propuestos en las instancias de la clase 6

Las Tablas A.7 a A.12 muestran los valores obtenidos al resolver las relajaciones de los modelos 1 y 2, así como el tiempo requerido para encontrar dicho resultado. Las filas en las cuales no se indica ningún valor de la cota corresponden a las instancias en las cuales el problema no pudo ser resuelto dentro de una hora.

Instancia	M1		M2	
	Cota	Tiempo (segs.)	Cota	Tiempo (segs.)
att48A	15.2979	76.221	17	0.051
att48B	-	>3600	17	0.03
att48C	9.7794	54.681	17	0.032
att48D	13.2732	61.206	17	0.032
att48E	14.9537	68.324	17	0.032
cmt121A	-	>3600	535	0.2
cmt121B	-	>3600	535	0.173
cmt121C	-	>3600	535	0.146
cmt121D	-	>3600	535	0.15
cmt151A	-	>3600	872	0.27
cmt151B	-	>3600	872	0.264
cmt151C	-	>3600	807.8802	0.312
cmt151D	-	>3600	830.3091	0.339
cmt151E	-	>3600	849.8789	0.311
cmt200A	-	>3600	1193.4673	0.719
cmt200B	-	>3600	1352	0.557
cmt200C	-	>3600	1221.6782	0.688
cmt200D	-	>3600	1265.2093	0.609

La tabla continúa en la página siguiente

Instancia	M1		M2	
	Cota	Tiempo (segs.)	Cota	Tiempo (segs.)
cmt200E	-	>3600	1295.2573	0.677
eil30A	6375	17.238	6375	0.014
eil30B	5326.8372	11.531	6375	0.013
eil30C	5878.9769	11.727	6375	0.013
eil30D	6275.138	13.639	6375	0.012
eil33A	5703.2993	10.942	14422.9425	0.018
eil33B	14417.7643	11.871	14530	0.016
eil33C	8363.9592	13.453	14477.096	0.018
eil33D	11976.7971	11.366	14530	0.018
eil33E	13140.0029	9.748	14530	0.017
eil51A	251.0908	82.855	287	0.036
eil51B	287	565.847	287	0.035
eil51C	125.8514	50.562	241.6735	0.037
eil51D	159.225	50.179	253.8794	0.038
eil51E	184.119	48.774	265.6685	0.038
eil76A	520.942	673.138	584.8833	0.077
eil76B	-	>3600	599	0.066
eil76C	239.1031	296.993	438.261	0.073
eil76D	316.2454	313.374	475.7831	0.058
eil76E	369.2137	369.341	507.2087	0.076
eil101A	-	>3600	612	0.118
eil101B	-	>3600	612	0.137
eil101C	287.874	2098.958	496.5024	0.157
eil101D	374.4767	2675.49	526.8079	0.182
eil101E	429.5125	3136.059	552.3258	0.165
gil262A	-	>3600	4948.6759	1.771
gil262B	-	>3600	4980	1.408
gil262C	-	>3600	4611.5867	1.754
gil262D	-	>3600	4750.2054	1.425
gil262E	-	>3600	4847.6267	1.574
op21A	178.8278	0.581	185	0.01
op21B	144.721	0.826	178.653	0.009
op21C	158.1621	0.766	182.8268	0.009
op21D	169.9276	0.641	185	0.01
op32A	94.5802	5.125	115	0.015
op32B	115	29.426	115	0.014
op32C	37.7273	4.397	102.7959	0.015
op32D	65.3659	3.461	109.6316	0.016
op32E	79.1574	4.978	115	0.016
op33A	267.7547	4.307	317.4809	0.016
op33B	338	4.897	340	0.015

La tabla continúa en la página siguiente

Instancia	M1		M2	
	Cota	Tiempo (segs.)	Cota	Tiempo (segs.)
op33C	113.6709	3.068	233.6682	0.017
op33D	172.7723	4.226	252.0548	0.017
op33E	192.5615	4.189	263.8814	0.015

Tabla A.7: Resultados obtenidos por Cplex 12.6 haciendo uso de la relajación lineal de los modelos propuestos en las instancias de la clase 1

Instancia	M1		M2	
	Cota	Tiempo (segs.)	Cota	Tiempo (segs.)
att48A	12	215.419	12	0.022
att48B	11.6674	121.207	12	0.021
att48C	12	204.47	12	0.021
att48D	12	248.821	12	0.022
cmt121A	-	>3600	330	0.152
cmt121B	-	>3600	330	0.156
cmt121C	-	>3600	330	0.152
cmt121D	-	>3600	330	0.14
cmt151A	-	>3600	561	0.242
cmt151B	-	>3600	561	0.215
cmt151C	-	>3600	557.5043	0.2
cmt151D	-	>3600	561	0.2
cmt151E	-	>3600	561	0.195
cmt200A	-	>3600	825	0.397
cmt200B	-	>3600	777.5502	0.369
cmt200C	-	>3600	807.028	0.475
cmt200D	-	>3600	822.7035	0.506
eil30A	3006.7656	4.144	3025	0.013
eil30B	1888.7932	2.444	3025	0.012
eil30C	2331.1705	4.098	3025	0.011
eil30D	2769.4825	4.414	3025	0.011
eil33A	8840.5072	4.174	9430	0.015
eil33B	8547.6829	4.422	9430	0.014
eil33C	9255.3416	5.625	9430	0.013
eil33D	9430	11.52	9430	0.014
eil51A	168	309.303	168	0.024
eil51B	153.2825	27.147	168	0.025
eil51C	168	68.83	168	0.025
eil51D	168	159.234	168	0.026
eil76A	386.8707	108.863	408.611	0.051
eil76B	-	>3600	409	0.042
eil76C	203.2161	83.151	358.3639	0.045
eil76D	299.3544	160.561	385.4661	0.044

La tabla continúa en la página siguiente

Instancia	M1		M2	
	Cota	Tiempo (segs.)	Cota	Tiempo (segs.)
eil76E	352.3138	84.247	402.8104	0.055
eil101A	567.7431	3099.807	408.8809	0.119
eil101B	-	>3600	409	0.101
eil101C	261.5187	475.413	402.9366	0.096
eil101D	327.0322	651.993	406.0227	0.12
eil101E	373.659	1508.944	408.124	0.103
gil262A	-	>3600	3003	1.035
gil262B	-	>3600	3003	0.7
gil262C	-	>3600	3003	0.701
gil262D	-	>3600	3003	0.778
op21A	74.2602	0.224	117.5516	0.009
op21B	131.3027	0.391	135	0.008
op21C	75.4011	0.247	118.0677	0.008
op21D	91.8717	0.338	125.4158	0.008
op21E	105.3211	0.304	129.6445	0.009
op32A	80	6.087	80	0.013
op32B	74.2366	4.059	80	0.015
op32C	80	7.348	80	0.012
op33A	212.3176	2.826	220	0.015
op33B	181.4504	2.441	220	0.014
op33C	216.0086	2.886	220	0.014
op33D	220	3.862	220	0.013

Tabla A.8: Resultados obtenidos por Cplex 12.6 haciendo uso de la relajación lineal de los modelos propuestos en las instancias de la clase 2

Instancia	M1		M2	
	Cota	Tiempo (segs.)	Cota	Tiempo (segs.)
att48A	4.6145	10.902	7	0.024
att48B	7	182.54	7	0.022
att48C	4.7846	10.284	7	0.023
att48D	5.5038	9.53	7	0.026
att48E	6.1288	6.1288	7	0.022
cmt121A	-	3600	152	0.106
cmt121B	-	3600	152	0.121
cmt121C	-	3600	152	0.126
cmt121D	-	3600	152	0.123
cmt151A	-	3600	199	0.172
cmt151B	-	3600	199	0.208
cmt151C	-	3600	199	0.203
cmt151D	-	3600	199	0.201
cmt200A	-	3600	463	0.363

La tabla continúa en la página siguiente

Instancia	M1		M2	
	Cota	Tiempo (segs.)	Cota	Tiempo (segs.)
cmt200B	-	3600	463	0.293
cmt200C	-	3600	463	0.373
cmt200D	-	3600	463	0.506
eil30A	961.815	5.809	2075	0.012
eil30B	2065.3407	4.304	2075	0.012
eil30C	2075	1.747	2075	0.011
eil30D	2075	1.874	2075	0.011
eil33A	5530	2.976	5530	0.013
eil33B	5530	2.455	5530	0.014
eil33C	5530	2.604	5530	0.014
eil51A	99	67.524	99	0.023
eil51B	96.4076	14.041	99	0.024
eil51C	99	41.05	99	0.024
eil51D	99	68.133	99	0.024
eil76A	168.153	44.125	202	0.045
eil76B	-	3600	202	0.036
eil76C	154.1259	40.589	202	0.035
eil76D	202	154.321	202	0.043
eil76E	202	2408.118	202	0.046
eil101A	-	3600	175	0.066
eil101B	-	3600	175	0.075
eil101C	-	3600	175	0.082
eil101D	-	3600	175	0.084
gil262A	-	3600	1369	0.743
gil262B	-	3600	1369	0.519
gil262C	-	3600	1369	0.554
gil262D	-	3600	1369	0.613
op21A	45	0.064	45	0.008
op21B	39.8	0.093	45	0.008
op21C	45	0.08	45	0.006
op21D	45	0.067	45	0.006
op32A	45	2.863	45	0.009
op32B	45	1.765	45	0.015
op32C	45	3.268	45	0.01
op32D	45	3.512	45	0.012
op33A	130	1.633	130	0.012
op33B	94.6535	1.009	130	0.012
op33C	128.2935	2.421	130	0.012
op33D	130	2.703	130	0.013

Tabla A.9: Resultados obtenidos por Cplex 12.6 haciendo uso de la relajación lineal de los modelos propuestos en las instancias de la clase 3

Instancia	M1		M2	
	Cota	Tiempo (segs.)	Cota	Tiempo (segs.)
att48A	21.4036	83.319	28.545	0.04
att48B	29.9455	123.703	33	0.034
att48C	13.0625	98.857	24.9729	0.037
att48D	17.4862	94.652	26.4204	0.034
att48E	20.275	78.033	27.7595	0.039
cmt121A	-	>3600	946	0.248
cmt121B	-	>3600	946	0.208
cmt121C	-	>3600	946	0.214
cmt121D	-	>3600	946	0.242
cmt151A	-	>3600	1390.4454	0.469
cmt151B	-	>3600	1553	0.448
cmt151C	-	>3600	1123.6491	0.479
cmt151D	-	>3600	1195.7109	0.406
cmt151E	-	>3600	1254.8305	0.408
cmt200A	-	>3600	1671.7342	0.881
cmt200B	-	>3600	2316.302	0.901
cmt200C	-	>3600	1727.7271	0.718
cmt200D	-	>3600	1835.3611	0.856
cmt200E	-	>3600	1922.6399	0.734
eil30A	9439.5555	13.539	10250	0.013
eil30B	5824.4068	8.408	9962.3976	0.012
eil30C	7540.4268	9.084	10079.0209	0.014
eil30D	8626.9629	14.862	10186.3279	0.012
eil33A	7235.9801	11.783	21005.9822	0.02
eil33B	20084.4852	13.18	21830	0.017
eil33C	10953.8037	10.02	21282.674	0.015
eil33D	15332.9575	14.198	21631.0041	0.02
eil33E	17754.6548	10.625	21810.6648	0.017
eil51A	338.4898	140.683	462.2865	0.042
eil51B	511.6458	172.34	534	0.042
eil51C	172.3887	49.266	321.8505	0.042
eil51D	210.6889	96.751	352.5749	0.04
eil51E	241.8105	79.029	382.9866	0.039
eil76A	637.5027	1009.562	792.9486	0.079
eil76B	928.3105	1347.695	988.7097	0.093
eil76C	280.2804	338.107	545.2851	0.061
eil76D	372.0296	478.236	593.2627	0.08
eil76E	446.3188	443.513	639.4757	0.083
eil101A	-	>3600	924.82	0.179
eil101B	-	>3600	1104.617	0.146
eil101C	-	>3600	611.6642	0.155

La tabla continúa en la página siguiente

Instancia	M1		M2	
	Cota	Tiempo (segs.)	Cota	Tiempo (segs.)
eil101D	-	>3600	665.6886	0.191
eil101E	-	>3600	712.8526	0.141
gil262A	-	>3600	7822.8155	1.729
gil262B	-	>3600	8293	1.297
gil262C	-	>3600	6769.1105	1.354
gil262D	-	>3600	7088.0136	1.688
gil262E	-	>3600	7360.1531	1.629
op21A	286.5129	1.286	375	0.008
op21B	189.5652	1.476	340.7033	0.01
op21C	228.6957	0.889	360.1811	0.008
op21D	263.7617	1.098	375	0.01
op32A	111.6949	3.005	161.5677	0.015
op32B	185.4408	5.436	214.882	0.016
op32C	38.2727	2.899	121.3053	0.012
op32D	77.8364	2.845	133.9274	0.012
op32E	94.1315	2.689	146.3783	0.012
op33A	325.0203	3.327	408.6839	0.013
op33B	472.9121	4.395	527.8006	0.014
op33C	126.4151	3.365	264.8457	0.014
op33D	188.4211	4.049	289.0199	0.013
op33E	225.6075	2.749	310.9896	0.014

Tabla A.10: Resultados obtenidos por Cplex 12.6 haciendo uso de la relajación lineal de los modelos propuestos en las instancias de la clase 4

Instancia	M1		M2	
	Cota	Tiempo (segs.)	Cota	Tiempo (segs.)
att48A	24.52	37.927	28	0.026
att48B	19.4416	52.124	26.7234	0.028
att48C	23.374	32.449	27.9998	0.029
att48D	25.3166	38.058	28	0.025
cmt121A	-	>3600	780	0.196
cmt121B	-	>3600	780	0.207
cmt121C	-	>3600	780	0.179
cmt121D	-	>3600	780	0.199
cmt151A	-	>3600	1094.9218	0.357
cmt151B	-	>3600	1279	0.252
cmt151C	-	>3600	1065.6919	0.304
cmt151D	-	>3600	1128.8214	0.314
cmt151E	-	>3600	1180.1396	0.273
cmt200A	-	>3600	1830.7049	0.717
cmt200B	-	>3600	1483.0458	0.806
cmt200C	-	>3600	1567.9871	0.669
cmt200D	-	>3600	1643.609	0.652
eil30A	6398.1022	9.969	7800	0.013
eil30B	3955.6016	6.675	7534.7427	0.013
eil30C	4981.2674	8.764	7651.7665	0.015
eil30D	5689.6848	9.901	7757.7358	0.016
eil33A	14284.9733	7.716	16880	0.017
eil33B	13347.2276	11.92	16880	0.016
eil33C	15020.7702	10.479	16880	0.016
eil33D	15846.0344	17.868	16880	0.015
eil51A	406.219	64.939	421	0.032
eil51B	252.2467	59.721	387.4718	0.038
eil51C	315.4904	61.015	406.4726	0.035
eil51D	360.2992	121.797	417.2258	0.034
eil76A	490.409	404.655	611.9851	0.078
eil76B	758.4111	1456.624	814.5264	0.079
eil76C	235.6542	206.071	450.8082	0.077
eil76D	348.9233	219.455	506.5399	0.08
eil76E	418.4608	205.026	559.8922	0.063
eil101A	567.7431	3099.807	725.1163	0.152
eil101B	299.8254	2877.602	885.928	0.17
eil101C	299.8254	1952.522	585.5589	0.115
eil101D	422.669	2078.692	643.8295	0.152
eil101E	522.6119	1763.1	699.766	0.153
gil262A	-	>3600	6719	1.395
gil262B	-	>3600	6429.0707	1.326

La tabla continúa en la página siguiente

Instancia	M1		M2	
	Cota	Tiempo (segs.)	Cota	Tiempo (segs.)
gil262C	-	>3600	6577.6889	1.357
gil262D	-	>3600	6679.4361	1.487
op21A	140.0571	0.727	264.3194	0.008
op21B	260.514	0.924	345	0.008
op21C	141.7186	0.675	266.1713	0.007
op21D	165.7061	0.64	291.5392	0.008
op21E	189.5898	0.612	313.8363	0.007
op32A	154.4275	3.958	184.882	0.021
op32B	118.6093	5.66	169.549	0.016
op32C	142.4797	4.149	179.1264	0.015
op33A	409.5676	7.676	460	0.017
op33B	292.6316	9.332	453.9479	0.019
op33C	418.8889	5.959	460	0.016
op33D	446.5772	8.926	460	0.016

Tabla A.11: Resultados obtenidos por Cplex 12.6 haciendo uso de la relajación lineal de los modelos propuestos en las instancias de la clase 5

Instancia	M1		M2	
	Cota	Tiempo (segs.)	Cota	Tiempo (segs.)
att48A	11.611	21.004	16.8298	0.033
att48B	20.0103	52.592	23	0.026
att48C	12.2826	25.105	17.0948	0.026
att48D	14.7233	32.213	18.7969	0.025
att48E	16.1211	30.798	20.3159	0.027
cmt121A	-	>3600	630	0.148
cmt121B	-	>3600	630	0.152
cmt121C	-	>3600	630	0.154
cmt121D	-	>3600	630	0.159
cmt151A	-	>3600	963	0.237
cmt151B	-	>3600	890.7872	0.283
cmt151C	-	>3600	937.8853	0.282
cmt151D	-	>3600	963	0.261
cmt200A	-	>3600	1591.619	0.668
cmt200B	-	>3600	1486.148	0.611
cmt200C	-	>3600	1546.59	0.835
cmt200D	-	>3600	1583.491	0.941
eil30A	1692.1433	7.81	5850	0.012
eil30B	4921.5349	9.101	5850	0.012
eil30C	5594.4062	11.95	5850	0.015
eil30D	5850	29.919	5850	0.011
eil33A	11207.7051	9.957	14030	0.019

La tabla continúa en la página siguiente

Instancia	M1		M2	
	Cota	Tiempo (segs.)	Cota	Tiempo (segs.)
eil33B	13548.1184	13.529	14030	0.015
eil33C	14030	38.282	14030	0.015
eil51A	332.4829	32.021	375	0.031
eil51B	213.481	23.481	368.2036	0.032
eil51C	289.5457	26.259	375	0.03
eil51D	349.8602	38.187	375	0.028
eil76A	249.3118	95.943	446.1704	0.073
eil76B	623.5957	1666.616	671	0.071
eil76C	221.2989	174.646	432.8493	0.074
eil76D	333.1925	227.048	495.0064	0.079
eil76E	406.723	559.468	551.8351	0.075
eil101A	635.9457	2395.055	675.5283	0.159
eil101B	355.8203	1027.96	541.2182	0.143
eil101C	478.8726	3072.011	598.8818	0.143
eil101D	557.7065	2753.881	644.6472	0.123
gil262A	-	>3600	5247	0.993
gil262B	-	>3600	5133.2675	1.082
gil262C	-	>3600	5238.7523	1.262
gil262D	-	>3600	5247	1.286
op21A	179.5105	0.83	259.759	0.009
op21B	85.1449	0.605	229.0126	0.011
op21C	124.4203	0.669	240.8434	0.009
op21D	186.8531	0.721	262.5455	0.011
op32A	116.3964	6.585	149.882	0.018
op32B	69.5455	3.567	136.7881	0.016
op32C	109.7594	6.074	147.4277	0.016
op32D	134.3902	4.662	150	0.015
op33A	359.0123	5.77	410	0.016
op33B	135.0442	7.521	338.4633	0.016
op33C	254.1117	4.336	371.6659	0.015
op33D	315.2475	2.462	400.5509	0.016

Tabla A.12: Resultados obtenidos por Cplex 12.6 haciendo uso de la relajación lineal de los modelos propuestos en las instancias de la clase 6

## A.2 TABLAS REFERENTES A LOS RESULTADOS DEL HÍBRIDO GRASP-VNS

Las Tablas A.13 a A.18 muestran los resultados obtenidos por el metaheurístico híbrido GRASP-VNS en las instancias contenidas en las clases 1 a 6, respectivamente.

En cada una de las tablas se presenta el valor de la función objetivo de la mejor y la peor soluciones encontradas por el algoritmo y el gap entre ellas, así como el tiempo en segundos que fue necesario para encontrar la mejor solución. Para cada una de las instancias para las cuales se conoce el valor óptimo, se expone el gap de la mejor y la peor solución con respecto al él.

Instancia	Mejor solución	Gap con óptimo (%)	Peor solución	Gap con óptimo (%)	Tiempo (segs.)	Gap entre soluciones
att48A	15	0	15	0	5.98	0
att48B	17	0	17	0	5.59	0
att48C	9	0	9	0	2.14	0
att48D	13	0	11	15.38	4.31	15.38
att48E	14	0	14	0	5.78	0
cmt121A	535	-	535	-	92.4783	0
cmt121B	416	-	406	-	121.79	2.4
cmt121C	501	-	493	-	168.62	1.6
cmt121D	530	-	530	-	113.09	0
cmt151A	809	-	800	-	398.04	1.11
cmt151B	872	0	872	0	76.93	0
cmt151C	434	-	417	-	240.7	3.92
cmt151D	558	-	540	-	366.34	3.23
cmt151E	664	-	658	-	306.7	0.9
cmt200A	595	-	542	-	506.73	8.91
cmt200B	1352	-	1352	-	236.59	0
cmt200C	745	-	696	-	489.88	6.58
cmt200D	891	-	869	-	628.11	2.47
cmt200E	1001	-	979	-	673.55	2.2
eil30A	6375	0	6375	0	5.03	0
eil30B	5125	0	5125	0	2.72	0
eil30C	5775	0	5775	0	5.46	0
eil30D	6275	0	6275	0	5.81	0
eil33A	5230	0	5230	0	5.94	0

La tabla continúa en la página siguiente

Instancia	Mejor solución	Gap con óptimo (%)	Peor solución	Gap con óptimo (%)	Tiempo (segs.)	Gap entre soluciones
eil33B	14380	0	14380	0	7.83	0
eil33C	7430	0	7430	0	7.61	0
eil33D	11630	0	11630	0	9.19	0
eil33E	12830	0	12830	0	9.59	0
eil51A	245	0	245	0	18.24	0
eil51B	287	0	287	0	11.35	0
eil51C	122	0	122	0	6.09	0
eil51D	150	0	148	1.33	12.98	1.33
eil51E	177	0	175	1.13	17.64	1.13
eil76A	520	0	513	1.35	39.56	1.35
eil76B	599	0	599	0	21.45	0
eil76C	232	0	232	0	25.92	0
eil76D	312	0	312	0	33.18	0
eil76E	367	-	364	-	35.2	0.82
eil101A	565	-	556	-	83.8	1.59
eil101B	612	0	612	0	31.83	0
eil101C	281	-	281	-	47.01	0
eil101D	367	-	355	-	64.68	3.27
eil101E	414	-	407	-	70.25	1.69
gil262A	4103	-	3995	-	1324.14	2.63
gil262B	4980	-	4980	-	440.55	0
gil262C	2502	-	2006	-	832.27	19.82
gil262D	3024	-	2853	-	883.44	5.65
gil262E	3555	-	2903	-	722.83	18.34
op21A	165	0	165	0	1.06	0
op21B	135	0	135	0	0.34	0
op21C	150	0	150	0	0.79	0
op21D	155	0	155	0	0.99	0
op32A	85	0	85	0	3.48	0
op32B	115	0	115	0	2.55	0
op32C	35	0	35	0	0.29	0
op32D	60	0	60	0	1.27	0
op32E	75	0	75	0	2.76	0
op33A	260	0	260	0	3.7	0
op33B	330	0	330	0	2.97	0
op33C	110	0	110	0	0.45	0
op33D	160	0	160	0	0.88	0
op33E	180	0	180	0	1.55	0

Tabla A.13: Resultados obtenidos el híbrido GRASP-VNS en las instancias de la clase 1

Instancia	Mejor solución	Gap con óptimo (%)	Peor solución	Gap con óptimo (%)	Tiempo (segs.)	Gap entre soluciones
att48A	12	0	12	0	1.46	0
att48B	11	0	11	0	2.79	0
att48C	12	0	12	0	1.52	0
att48D	12	0	12	0	1.46	0
cmt121A	330	-	327	-	116.98	0.91
cmt121B	305	-	292	-	196.69	4.26
cmt121C	329	-	327	-	130.26	0.61
cmt121D	330	-	330	-	52.11	0
cmt151A	460	-	404	-	108.41	12.17
cmt151B	561	-	561	-	46.26	0
cmt151C	403	-	320	-	203.12	20.6
cmt151D	496	-	465	-	211.32	6.25
cmt151E	534	-	515	-	173.95	3.56
cmt200A	825	-	825	-	147.87	0
cmt200B	481	-	368	-	449.16	23.49
cmt200C	591	-	559	-	432.96	5.41
cmt200D	694	-	675	-	460.67	2.74
eil30A	2925	0	2925	0	1.39	0
eil30B	1625	0	1625	0	0.59	0
eil30C	1625	0	1625	0	0.61	0
eil30D	2475	0	2475	0	0.82	0
eil33A	8480	0	8480	0	3.71	0
eil33B	8480	0	8480	0	3.95	0
eil33C	8480	0	8480	0	3.7	0
eil33D	9430	0	9430	0	3.31	0
eil51A	168	0	168	0	6.42	0
eil51B	148	0	148	0	11.42	0
eil51C	168	0	168	0	10.29	0
eil51D	168	0	168	0	8.18	0
eil76A	377	-	377	-	22.67	0
eil76B	409	0	409	0	9.65	0
eil76C	201	0	194	3.48	26.79	3.48
eil76D	294	0	294	0	21.39	0
eil76E	345	0	339	1.74	25.56	1.74
eil101A	383	-	375	-	51.21	2.09
eil101B	409	0	409	0	21.4	0
eil101C	260	-	251	-	100.57	3.46
eil101D	326	-	326	-	56.26	0
eil101E	370	-	359	-	109.54	2.97
gil262A	3003	-	3003	-	213.06	0

La tabla continúa en la página siguiente

Instancia	Mejor solución	Gap con óptimo (%)	Peor solución	Gap con óptimo (%)	Tiempo (segs.)	Gap entre soluciones
gil262B	2038	-	1789	-	768.49	12.22
gil262C	2759	-	2481	-	694.05	10.08
gil262D	2956	-	2921	-	586.31	1.18
op21A	60	0	60	0	0.09	0
op21B	125	0	125	0	0.18	0
op21C	60	0	60	0	0.09	0
op21D	60	20	60	20	0.09	0
op21E	80	0	80	0	0.12	0
op32A	80	0	80	0	0.81	0
op32B	70	0	70	0	1.21	0
op32C	80	0	80	0	1	0
op33A	210	0	210	0	1.04	0
op33B	170	0	170	0	0.86	0
op33C	210	0	210	0	1.02	0
op33D	220	0	220	0	0.88	0

Tabla A.14: Resultados obtenidos el híbrido GRASP-VNS en las instancias de la clase 2

Instancia	Mejor solución	Gap con óptimo (%)	Peor solución	Gap con óptimo (%)	Tiempo (segs.)	Gap entre soluciones
att48A	4	0	4	0	2.09	0
att48B	7	0	7	0	1.47	0
att48C	4	0	4	0	0.4	0
att48D	5	0	5	0	2.57	0
att48E	5	0	5	0	2.18	0
cmt121A	152	-	152	-	43.26	0
cmt121B	152	-	152	-	47.6	0
cmt121C	152	-	152	-	39.13	0
cmt121D	152	-	152	-	28.74	0
cmt151A	199	-	199	-	24.66	0
cmt151B	171	-	113	-	87.21	33.92
cmt151C	199	-	199	-	37.11	0
cmt151D	199	-	199	-	26.69	0
cmt200A	463	-	463	-	65.98	0
cmt200B	354	-	292	-	218.79	17.51
cmt200C	438	-	431	-	179.75	1.6
cmt200D	463	-	463	-	78.9	0
eil30A	950	0	950	0	0.67	0
eil30B	1950	0	1950	0	0.6	0

La tabla continúa en la página siguiente

Instancia	Mejor solución	Gap con óptimo (%)	Peor solución	Gap con óptimo (%)	Tiempo (segs.)	Gap entre soluciones
eil30C	2075	0	2075	0	0.64	0
eil30D	2075	0	2075	0	0.55	0
eil33A	5530	0	5530	0	0.82	0
eil33B	5530	0	5530	0	0.61	0
eil33C	5530	0	5530	0	0.57	0
eil51A	99	0	99	0	2.67	0
eil51B	90	0	85	5.56	4.87	5.56
eil51C	99	0	99	0	3.92	0
eil51D	99	0	99	0	2.56	0
eil76A	167	0	97	41.92	24.31	41.92
eil76B	202	0	202	0	4.24	0
eil76C	142	0	139	2.11	18.86	2.11
eil76D	193	-	193	-	15.2	0
eil76E	202	0	202	0	7.83	0
eil101A	175	0	175	0	7.96	0
eil101B	155	-	137	-	40.25	11.61
eil101C	175	-	174	-	15.75	0.57
eil101D	175	0	175	0	11.05	0
gil262A	1369	-	1369	-	115.73	0
gil262B	1147	-	665	-	578.91	42.02
gil262C	1366	-	1317	-	346.03	3.59
gil262D	1369	-	1369	-	128.05	0
op21A	45	0	45	0	0.04	0
op21B	30	0	30	0	0.04	0
op21C	45	0	45	0	0.04	0
op21D	45	0	45	0	0.04	0
op32A	45	0	45	0	0.04	0
op32B	45	0	45	0	0.51	0
op32C	45	0	45	0	0.54	0
op32D	45	0	45	0	0.38	0
op33A	130	0	130	0	0.18	0
op33B	80	0	80	0	0.21	0
op33C	120	0	120	0	0.24	0
op33D	130	0	130	0	0.23	0

Tabla A.15: Resultados obtenidos el híbrido GRASP-VNS en las instancias de la clase 3

Instancia	Mejor solución	Gap con óptimo (%)	Peor solución	Gap con óptimo (%)	Tiempo (segs.)	Gap entre soluciones
att48A	21	0	20	4.76	7.56	4.76
att48B	28	3.45	28	3.45	9.74	0
att48C	13	0	12	7.69	3.45	7.69
att48D	17	0	16	5.88	6.09	5.88
att48E	20	0	19	5	7.42	5
cmt121A	803	-	791	-	465.44	1.49
cmt121B	512	-	498	-	257.09	2.73
cmt121C	644	-	632	-	412.98	1.86
cmt121D	751	-	737	-	381.74	1.86
cmt151A	1048	-	1018	-	620.91	2.86
cmt151B	1446	-	1434	-	349.43	0.83
cmt151C	469	-	440	-	306.97	6.18
cmt151D	629	-	609	-	326.59	3.18
cmt151E	769	-	740	-	358.02	3.77
cmt200A	723	-	630	-	594.27	12.86
cmt200B	2097	-	2067	-	1770.94	1.43
cmt200C	869	-	804	-	646.46	7.48
cmt200D	1066	-	1031	-	802.44	3.28
cmt200E	1237	-	1204	-	662.52	2.67
eil30A	9375	0	9375	0	10.02	0
eil30B	5450	0	5450	0	3.54	0
eil30C	7300	0	7300	0	8.93	0
eil30D	8575	0	8575	0	8.34	0
eil33A	6950	0	6950	0	7.39	0
eil33B	19730	0	19730	0	14.09	0
eil33C	10630	0	10630	0	10.88	0
eil33D	14180	0	14180	0	11.58	0
eil33E	17430	0	17430	0	16.01	0
eil51A	328	0	325	0.91	47.64	0.91
eil51B	509	0	509	0	30.12	0
eil51C	168	0	168	0	9.62	0
eil51D	201	0	200	0.5	20.03	0.5
eil51E	232	0	232	0	21.3	0
eil76A	616	-	605	-	118	1.79
eil76B	919	0.43	903	2.17	130.42	1.74
eil76C	274	-	274	-	35.15	0
eil76D	366	-	366	-	46.38	0
eil76E	442	-	435	-	63.91	1.58
eil101A	774	-	761	-	233.44	1.68
eil101B	1057	-	1050	-	221.92	0.66

La tabla continúa en la página siguiente

Instancia	Mejor solución	Gap con óptimo (%)	Peor solución	Gap con óptimo (%)	Tiempo (segs.)	Gap entre soluciones
eil101C	326	-	326	-	84.38	0
eil101D	422	-	404	-	150.17	4.27
eil101E	489	-	445	-	108.25	9
gil262A	5276	-	4911	-	1974.23	6.92
gil262B	7665	-	7551	-	2931.03	1.49
gil262C	2842	-	2413	-	976.81	15.1
gil262D	3714	-	2771	-	1087.86	25.39
gil262E	4062	-	3851	-	1450.23	5.19
op21A	260	0	260	0	2.63	0
op21B	180	0	180	0	1.08	0
op21C	215	0	215	0	2.3	0
op21D	260	0	260	0	2.44	0
op32A	110	0	110	0	3.93	0
op32B	180	0	180	0	3.23	0
op32C	35	0	35	0	0.3	0
op32D	75	0	75	0	1.65	0
op32E	90	0	90	0	2.78	0
op33A	310	0	310	0	6.54	0
op33B	460	0	460	0	6.46	0
op33C	120	0	120	0	0.61	0
op33D	180	0	180	0	1.45	0
op33E	220	0	220	0	2.86	0

Tabla A.16: Resultados obtenidos el híbrido GRASP-VNS en las instancias de la clase 4

Instancia	Mejor solución	Gap con óptimo (%)	Peor solución	Gap con óptimo (%)	Tiempo (segs.)	Gap entre soluciones
att48A	24	0	24	0	6.37	0
att48B	19	0	19	0	5.88	0
att48C	23	0	23	0	6.64	0
att48D	25	0	25	0	6.41	0
cmt121A	621	-	587	-	336.34	5.48
cmt121B	444	-	411	-	368.46	7.43
cmt121C	615	-	582	-	409.95	5.37
cmt121D	685	-	673	-	240.98	1.75
cmt151A	657	-	581	-	359.49	11.57
cmt151B	1168	-	1157	-	467.02	0.94
cmt151C	584	-	438	-	368.19	25
cmt151D	735	-	695	-	280.49	5.44

La tabla continúa en la página siguiente

Instancia	Mejor solución	Gap con óptimo (%)	Peor solución	Gap con óptimo (%)	Tiempo (segs.)	Gap entre soluciones
cmt151E	861	-	803	-	470.7	6.74
cmt200A	1615	-	1581	-	858.56	2.11
cmt200B	636	-	588	-	880.79	7.55
cmt200C	901	-	821	-	737.14	8.88
cmt200D	1141	-	1058	-	693.69	7.27
eil30A	6225	0	6225	0	8.52	0
eil30B	3350	0	3350	0	3.51	0
eil30C	4750	0	4750	0	3.86	0
eil30D	5075	0	5075	0	7.64	0
eil33A	14230	0	14230	0	9.97	0
eil33B	12880	0	12880	0	9.4	0
eil33C	14930	0	14930	0	9.81	0
eil33D	15680	0	15680	0	7.52	0
eil51A	402	0	402	0	21.73	0
eil51B	235	0.42	225	4.66	21.71	4.26
eil51C	302	0	295	2.32	27.02	2.32
eil51D	357	0	347	2.8	28.97	2.8
eil76A	481	-	465	-	56.98	3.33
eil76B	745	-	731	-	52.06	1.88
eil76C	228	-	183	-	49.49	19.74
eil76D	341	-	324	-	62.67	4.99
eil76E	411	-	394	-	76.48	4.14
eil101A	546	-	517	-	171.98	5.31
eil101B	854	-	838	-	178.14	1.87
eil101C	296	-	183	-	133.14	38.18
eil101D	409	-	385	-	110.45	5.87
eil101E	485	-	468	-	160.06	3.51
gil262A	6079	-	5999	-	1913.11	1.32
gil262B	2974	-	2093	-	1566.06	29.62
gil262C	4244	-	3693	-	1679.43	12.98
gil262D	5039	-	4842	-	4899.67	3.91
op21A	135	0	135	0	0.17	0
op21B	235	0	235	0	1.09	0
op21C	135	0	135	0	0.18	0
op21D	135	0	135	0	0.17	0
op21E	165	0	165	0	0.27	0
op32A	150	0	150	0	3.05	0
op32B	110	0	110	0	2.65	0
op32C	140	0	140	0	3.05	0
op33A	400	0	400	0	8.35	0

La tabla continúa en la página siguiente

Instancia	Mejor solución	Gap con óptimo (%)	Peor solución	Gap con óptimo (%)	Tiempo (segs.)	Gap entre soluciones
op33B	270	0	270	0	2.85	0
op33C	410	0	410	0	7.91	0
op33D	440	0	440	0	7.25	0

Tabla A.17: Resultados obtenidos el híbrido GRASP-VNS en las instancias de la clase 5

Instancia	Mejor solución	Gap con óptimo (%)	Peor solución	Gap con óptimo (%)	Tiempo (segs.)	Gap entre soluciones
att48A	11	0	11	0	5.35	0
att48B	19	0	18	5.26	8.7	5.26
att48C	12	0	12	0	4.9	0
att48D	14	0	14	0	6.46	0
att48E	15	6.25	15	6.25	7.51	0
cmt121A	473	-	460	-	348.92	2.75
cmt121B	429	-	411	-	360.05	4.2
cmt121C	540	-	521	-	379.92	3.52
cmt121D	608	-	599	-	263	1.48
cmt151A	838	-	830	-	465.13	0.95
cmt151B	488	-	Infactible	-	413.18	-
cmt151C	702	-	670	-	458.7	4.56
cmt151D	824	-	806	-	465.22	2.18
cmt200A	1358	-	1331	-	639.65	1.99
cmt200B	792	-	653	-	773.02	17.55
cmt200C	1147	-	1039	-	791.76	9.42
cmt200D	1327	-	1224	-	801.21	7.76
eil30A	1550	0	1550	0	4.07	0
eil30B	4775	0	4775	0	4.04	0
eil30C	5550	0	5550	0	2.87	0
eil30D	5850	0	5725	2.14	2.08	2.14
eil33A	10600	0	10600	0	5.37	0
eil33B	13330	0	13330	0	4.48	0
eil33C	14030	0	14030	0	2.18	0
eil51A	320	2.74	309	6.08	29.32	3.44
eil51B	212	0	196	7.55	18.08	7.55
eil51C	272	5.88	262	9.34	23.4	3.68
eil51D	343	0.29	329	4.36	26.01	4.08
eil76A	229	-	228	-	59.52	0.44
eil76B	599	-	597	-	37.94	0.33
eil76C	211	0	205	2.84	60.5	2.84

La tabla continúa en la página siguiente

Instancia	Mejor solución	Gap con óptimo (%)	Peor solución	Gap con óptimo (%)	Tiempo (segs.)	Gap entre soluciones
eil76D	304	-	284	-	51.58	6.58
eil76E	380	-	371	-	47.48	2.37
eil101A	614	-	610	-	163.29	0.65
eil101B	337	-	241	-	155.74	28.49
eil101C	460	-	425	-	161.93	7.61
eil101D	552	-	526	-	166.03	4.71
gil262A	4444	-	4203	-	944.43	5.42
gil262B	2493	-	1800	-	1862.28	27.8
gil262C	3660	-	3238	-	1124.5	11.53
gil262D	4352	-	4116	-	1215.46	5.42
op21A	155	0	155	0	0.76	0
op21B	75	0	75	0	0.21	0
op21C	110	0	110	0	0.5	0
op21D	160	0	155	3.13	0.9	3.13
op32A	115	0	115	0	3.1	0
op32B	65	0	65	0	3.31	0
op32C	105	0	105	0	2.97	0
op32D	130	0	130	0	2.72	0
op33A	350	0	350	0	2.82	0
op33B	130	0	130	0	2.6	0
op33C	240	0	240	0	3.24	0
op33D	300	0	300	0	3.37	0

Tabla A.18: Resultados obtenidos el híbrido GRASP-VNS en las instancias de la clase 6

### A.3 TABLAS REFERENTES A LOS RESULTADOS DEL MÉTODO DE GENERACIÓN DE COLUMNAS

En las Tablas A.19 a A.24 se muestran las cotas superior (CS) e inferior (CI) obtenidas por el método de generación de columnas, así como el gap con respecto al óptimo de cada una y el gap entre ellas. Además se presenta el número de iteraciones requeridas (#I) y el número de columnas generadas (#C) por el método. Por último, se muestra el tiempo de CPU que fue requerido para obtener dichas cotas.

Instancia	CS	CI	Gap entre cotas	Gap (CS)	Gap (CI)	#I	#C	Tiempo (segs.)
att48A	15.6258	13	16.8	4	13.33	42	721	6752.47
att48B	17	17	0	0	0	22	510	0.595312
att48C	9.75	8	17.95	7.69	11.11	36	400	21717.3
att48D	14.8333	11	25.84	12.36	15.38	15	378	11393
att48E	18.4258	12	34.87	24.02	14.29	33	502	3863.83
eil30A	7596.21	6325	16.73	16.08	0.78	45	552	0.339743
eil30B	5125	5125	0	0	0	18	383	6821.33
eil30C	5900	5725	2.97	2.12	0.87	37	601	632.265
eil30D	6275	5975	4.78	0	4.78	41	803	439.482
eil33A	5546.67	5230	5.71	5.71	0	24	265	6211.15
eil33B	14755.9	13480	8.65	2.55	6.26	50	1261	0.946404
eil33C	7680	7430	3.26	3.26	0	26	299	2770.43
eil33D	11920	11030	7.47	2.43	5.16	30	570	214.863
eil33E	13255	12830	3.21	3.21	0	40	1045	311.183
op21A	168.333	165	1.98	1.98	0	23	162	48.8425
op21B	135	135	0	0	0	18	181	91.2364
op21C	150	150	0	0	0	28	278	26.2029
op21D	155	150	3.23	0	3.23	17	190	53.6015
op32A	96	85	11.46	11.46	0	35	352	1228.84
op32B	151.406	110	27.35	24.05	4.35	49	489	0.391554
op32C	35	35	0	0	0	30	194	1388.33
op32D	62.5	60	4	4	0	48	363	2201.15
op32E	78.75	75	4.76	4.76	0	56	445	4793.16
op33A	264.545	240	9.28	1.72	7.69	29	553	241.884
op33B	343.182	310	9.67	3.84	6.06	68	1031	1.32638
op33C	110	110	0	0	0	26	217	1191.07
op33D	160	160	0	0	0	42	324	2374.01

La tabla continúa en la página siguiente

Instancia	CS	CI	Gap entre cotas	Gap (CS)	Gap (CI)	#I	#C	Tiempo (segs.)
op33E	190	180	5.26	5.26	0	27	326	438.552

Tabla A.19: Resultados obtenidos por el método de generación de columnas en las instancias de la clase 1

Instancia	CS	CI	Gap entre cotas	Gap (CS)	Gap (CI)	#I	#C	Tiempo (segs.)
att48A	12	12	0	0	0	39	936	1.81617
att48B	12.9416	11	15	15	0	87	1392	22474.1
att48C	12	12	0	0	0	51	1075	3.51959
att48D	12	12	0	0	0	76	828	3.33597
eil30A	3050	2925	4.1	4.1	0	39	687	0.786312
eil30B	1650	1625	1.52	1.52	0	22	476	1576.52
eil30C	1725	1625	5.8	5.8	0	43	728	6226.29
eil30D	2712.5	2475	8.76	8.76	0	39	710	336.655
eil33A	8780	8480	3.42	3.42	0	99	2285	438.641
eil33B	8600	8480	1.4	1.4	0	80	2085	549.117
eil33C	8650	8480	1.97	1.97	0	147	2680	1472.94
eil33D	9430	9430	0	0	0	177	550	1.36051
op21A	60	60	0	0	0	19	152	52.4341
op21B	125	100	20	20	0	44	348	48.3976
op21C	60	60	0	0	0	31	185	37.0803
op21D	75	75	0	0	0	30	227	41.5493
op21E	80	80	0	0	0	31	259	236.911
op32A	80	80	0	0	0	95	794	1.11756
op32B	70	65	7.14	7.14	0	50	676	765.348
op32C	80	80	0	0	0	88	922	1.36191
op33A	212.857	210	1.34	1.34	0	182	2270	1485.38
op33B	170	170	0	0	0	57	953	1637.01
op33C	231.429	200	13.58	13.58	0	196	2451	3686.61
op33D	220	220	0	0	0	189	637	2.47018

Tabla A.20: Resultados obtenidos por el método de generación de columnas en las instancias de la clase 2

Instancia	CS	CI	Gap entre cotas	Gap (CS)	Gap (CI)	#I	#C	Tiempo (segs.)
att48A	9.8695	4	59.47	59.47	0	43	826	22313.6

La tabla continúa en la página siguiente

Instancia	CS	CI	Gap entre cotas	Gap (CS)	Gap (CI)	#I	#C	Tiempo (segs.)
att48B	7	7	0	0	0	42	605	1.08355
att48C	11.0456	4	63.79	63.79	0	38	896	21292.1
att48D	11.6431	5	57.06	57.06	0	45	994	21158.3
att48E	9.67121	5	48.3	48.3	0	59	1408	47240.4
eil30A	950	950	0	0	0	60	1090	5105.94
eil30B	2075	1950	6.02	6.02	0	126	1079	1.2864
eil30C	2075	2075	0	0	0	203	403	0.688118
eil30D	2144.92	2075	3.26	3.26	0	179	209	0.382729
eil33A	5745.64	5530	3.75	3.75	0	141	458	0.544007
eil33B	5530	5530	0	0	0	252	285	0.790886
eil33C	5530	5530	0	0	0	254	284	0.819646
op21A	45	45	0	0	0	33	210	0.094177
op21B	30	30	0	0	0	30	262	66.1241
op21C	45	45	0	0	0	25	217	0.082073
op21D	45	45	0	0	0	47	193	0.130378
op32A	45	45	0	0	0	92	409	0.385304
op32B	45	45	0	0	0	44	555	0.407688
op32C	45	45	0	0	0	68	337	0.265175
op32D	45	45	0	0	0	114	554	0.606851
op33A	130	130	0	0	0	154	1331	2.50939
op33B	86.6667	80	7.69	7.69	0	81	1004	6710.22
op33C	125	120	4	4	0	137	2428	477.509
op33D	130	130	0	0	0	123	1835	3.89369

Tabla A.21: Resultados obtenidos por el método de generación de columnas en las instancias de la clase 3

Instancia	CS	CI	Gap entre cotas	Gap (CS)	Gap (CI)	#I	#C	Tiempo (segs.)
att48A	22	19	13.64	4.55	9.52	23	327	4591.12
att48B	31.1667	27	13.37	6.95	6.9	43	674	11077.9
att48C	14	9	35.71	7.14	30.77	14	259	5613.58
att48D	20	14	30	15	17.65	13	265	38334.7
att48E	21.75	17	21.84	8.05	15	23	284	17252.4
eil30A	9560.71	9075	5.08	1.94	3.2	30	396	500.16
eil30B	5650	5450	3.54	3.54	0	8	120	784.831
eil30C	7625	7300	4.26	4.26	0	8	156	1861.96
eil30D	8575	8575	0	0	0	17	159	281.439
eil33A	7810	6920	11.4	11.01	0.43	8	127	5570.06
eil33B	26770	19730	26.3	26.3	0	53	248	3656.75

La tabla continúa en la página siguiente

Instancia	CS	CI	Gap entre cotas	Gap (CS)	Gap (CI)	#I	#C	Tiempo (segs.)
eil33C	11030	10630	3.63	3.63	0	8	124	6140.6
eil33D	26770	13830	48.34	47.03	2.47	27	197	3795.65
eil33E	17490	16880	3.49	0.34	3.16	10	183	239.639
op21A	270	260	3.7	3.7	0	15	104	33.185
op21B	180	180	0	0	0	10	72	36.1699
op21C	220	215	2.27	2.27	0	13	91	105.25
op21D	260	260	0	0	0	4	59	135.669
op32A	260	110	57.69	57.69	0	32	124	3680.94
op32B	260	175	32.69	30.77	2.78	43	89	3632.55
op32C	35	35	0	0	0	15	103	1104.01
op32D	75	75	0	0	0	10	89	576.426
op32E	90	90	0	0	0	19	129	1350.31
op33A	320	310	3.13	3.13	0	16	233	412.967
op33B	472.5	460	2.65	2.65	0	46	785	1379
op33C	120	120	0	0	0	17	140	984.496
op33D	185	180	2.7	2.7	0	17	179	527.073
op33E	230	220	4.35	4.35	0	7	107	353.301

Tabla A.22: Resultados obtenidos por el método de generación de columnas en las instancias de la clase 4

Instancia	CS	CI	Gap entre cotas	Gap (CS)	Gap (CI)	#I	#C	Tiempo (segs.)
att48A	28.6045	24	16.1	16.1	0	50	1124	26885
att48B	23.7857	19	20.12	20.12	0	35	644	32090.4
att48C	27.521	23	16.43	16.43	0	46	1067	22200.9
att48D	27.6341	25	9.53	9.53	0	108	1501	20662.4
eil30A	9750	5375	44.87	36.15	13.65	76	241	3646.21
eil30B	9750	3350	65.64	65.64	0	27	203	3748.43
eil30C	9750	4700	51.79	51.28	1.05	29	234	3760.25
eil30D	9750	4875	50	47.95	3.94	50	95	3644.08
eil33A	16660	14230	14.59	14.59	0	67	334	3656.41
eil33B	25120	12500	50.24	48.73	2.95	55	75	3644.55
eil33C	17871.7	14930	16.46	16.46	0	69	328	3613.93
eil33D	18410	15680	14.83	14.83	0	61	299	3645.14
op21A	135	135	0	0	0	10	89	30.8779
op21B	235	225	4.26	0	4.26	19	162	35.4823
op21C	135	135	0	0	0	10	67	37.6295
op21D	135	135	0	0	0	19	136	124.593
op21E	170	165	2.94	2.94	0	11	66	113.791
op32A	230	150	34.78	34.78	0	15	203	3736.39
op32B	120	110	8.33	8.33	0	38	323	3899.52
op32C	230	140	39.13	39.13	0	31	289	3673.67
op33A	409.5	380	7.2	2.32	5	63	808	3714.53
op33B	285	250	12.28	5.26	7.41	33	356	3758.41
op33C	422.5	410	2.96	2.96	0	75	1132	93.0099
op33D	450	430	4.44	2.22	2.27	67	1021	84.4179

Tabla A.23: Resultados obtenidos por el método de generación de columnas en las instancias de la clase 5

Instancia	CS	CI	Gap entre cotas	Gap (CS)	Gap (CI)	#I	#C	Tiempo (segs.)
att48A	16.7468	11	34.32	34.32	0	32	534	29090.7
att48B	22.3216	18	19.36	14.88	5.26	77	1269	35405.1
att48C	18.4615	11	40.42	35	8.33	18	404	27496.4
att48D	17.6539	14	20.7	20.7	0	44	829	15983.7
att48E	19.5	15	23.08	17.95	6.25	62	949	4865.82
eil30A	2275	1550	31.87	31.87	0	27	320	6552.26
eil30B	4900	4775	2.55	2.55	0	48	760	1625.66
eil30C	5667.86	5550	2.08	2.08	0	96	1909	489.302
eil30D	5850	5700	2.56	0	2.56	14	39	0.049303
eil33A	10630	10080	5.17	0.28	4.91	71	153	3606.35

La tabla continúa en la página siguiente

Instancia	CS	CI	Gap entre cotas	Gap (CS)	Gap (CI)	#I	#C	Tiempo (segs.)
eil33B	15580	13330	14.44	14.44	0	57	55	3624.5
eil33C	14030	14030	0	0	0	60	106	0.170293
op21A	160	155	3.13	3.13	0	18	164	204.445
op21B	75	75	0	0	0	24	112	355.503
op21C	115	110	4.35	4.35	0	17	123	505.76
op21D	165	160	3.03	3.03	0	28	195	128.71
op32A	121.875	110	9.74	5.64	4.35	62	55	3645.3
op32B	105	55	47.62	38.1	15.38	65	22	3653.21
op32C	105	100	4.76	0	4.76	3	67	0.050553
op32D	130	130	0	0	0	4	35	0.041144
op33A	371.154	350	5.7	5.7	0	107	1619	3647.44
op33B	158.75	130	18.11	18.11	0	42	471	4611.08
op33C	250	220	12	4	8.33	69	643	3942.57
op33D	321.596	300	6.72	6.72	0	68	1206	6386.21

Tabla A.24: Resultados obtenidos por el método de generación de columnas en las instancias de la clase 6

## APÉNDICE B

# GENERACIÓN DE INSTANCIAS

---

Para realizar los experimentos computacionales fue necesaria la generación de instancias de prueba, debido a que la versión del Orienteering Problem que se propone en esta tesis no ha sido estudiada con anterioridad en la literatura.

Seis conjuntos de instancias de prueba fueron generados y estos están basados en las instancias de la Clase 1 de Fischetti *et al.* (1998) para el OP, las cuales a su vez, se basan en instancias del VRP capacitado. La demanda de cada cliente del VRP capacitado es vista como la puntuación del mismo en el OP y la duración máxima de la ruta es igual a  $\beta v(TSP)$ , donde  $v(TSP)$  es igual al valor de la función objetivo del TSP resuelto en ese grafo y  $\beta = 0.25, 0.5, 0.75$ . Los grafos contenidos en esta clase tienen dimensiones que van desde los 21 hasta los 262 nodos.

Para adaptar las instancias de Fischetti *et al.* (1998), se establecieron nodos obligatorios, se determinaron los conflictos entre nodos y se determinó una duración máxima del recorrido apropiada. Cada uno de los seis conjuntos de instancias está determinado por el mismo conjunto de grafos, variando el porcentaje de nodos obligatorios y el nivel de conflictos.

## B.1 GENERACIÓN DE CONFLICTOS EN NIVEL ALTO

Para generar un grafo con un nivel de conflictos alto se toma como entrada una instancia de la Clase 1 de Fischetti *et al.* (1998), la cual está denotada por

$I_f = (V_f, t_f, s_f)$ , donde  $V_f$  representa al conjunto de nodos,  $t_f$  a la matriz de tiempos de viaje internodales y  $s_f$  al vector de las puntuaciones asociadas a cada nodo.

El procedimiento de fijación de conflictos consiste en formar una lista  $L$  de nodos que, en principio, es igual al conjunto de nodos de  $V$ , menos los depósitos. Se elige un elemento  $k$  en  $L$  y se calcula un número aleatorio  $c_k \in \{0, 1, 2, \text{ y } 3\}$ . Si  $c_k = 0$ , el nodo es libre de incompatibilidades y se elimina de  $L$ ; si no, se eligen  $c_k$  nodos aleatorios en  $L$  y estos, conformarán el conjunto de nodos que tienen conflicto con  $k$  y serán removidos. Se considera que los conflictos cumplen las propiedades simétrica y transitiva y el proceso continúa hasta que la lista queda vacía.

El Algoritmo 11 muestra la manera en que los conflictos en nivel alto son fijados en un grafo.

---

**Algoritmo 11** Fijación de conflictos en nivel alto

---

**Entrada:**  $I_f = (V_f, t_f, s_f)$  : Instancia de la clase 1 de Fischetti para el OP

- 1:  $L \leftarrow V_f \setminus \{1, n\}$
- 2: **mientras**  $L \neq \emptyset$  **hacer**
- 3:   Seleccionar  $k \in L$
- 4:   Calcular  $c_k \in \{0, 1, 2, 3\}$  aleatoriamente
- 5:   **si**  $c_k=0$  , **entonces**
- 6:      $L \leftarrow L \setminus \{k\}$
- 7:   **si no**
- 8:     **si**  $|L| - 1 \geq c_k$  , **entonces**
- 9:       Seleccionar  $k_1, \dots, k_{c_k}$  nodos aleatorios en  $L$
- 10:        $C_k \leftarrow \{k_1, \dots, k_{c_k}\}$
- 11:     **si no**
- 12:        $C_k \leftarrow L \setminus \{k\}$
- 13:     **fin si**
- 14:      $C_i \leftarrow C_k \cup \{k\} \setminus \{i\}, \forall i \in C_k$
- 15:      $L \leftarrow L \setminus (C_k \cup \{k\})$
- 16:   **fin si**
- 17: **fin mientras**

**Salida:**  $C_k, k \in V_f \setminus \{1, n\}$  : Conjuntos de conflictos de instancias de clases 1, 2 y 3

---

## B.2 GENERACIÓN DE CONFLICTOS EN NIVEL BAJO

Para crear una instancia con un nivel bajo de incompatibilidades se toma como entrada una instancia que presente un nivel de conflictos alto, generada por el procedimiento descrito en la sección anterior. Posteriormente, se crea una lista  $L$  que contenga al conjunto de vértices, se calcula  $n = \lfloor |V_f|/2 \rfloor$  y se fija el número de conflictos removidos  $j = 0$ .

Mientras que  $j$  sea menor que  $n$  se selecciona un nodo aleatorio  $k$  que forme parte de  $L$ . Luego, cada nodo perteneciente a  $C_k \cup \{k\}$  se define como libre de conflicto y se elimina de  $L$ ; finalmente, se actualiza  $j = j + |C_k| + 1$ .

El procedimiento anteriormente descrito, se muestra de manera detallada en el Algoritmo 12.

---

### Algoritmo 12 Fijación de conflictos en nivel bajo

---

**Entrada:**  $C_k, k \in V_f \setminus \{1, n\}$  : Conjuntos de conflictos de instancias de clases 1, 2 y 3

1:  $n \leftarrow \lfloor |V_f|/2 \rfloor$       \Número de conflictos a remover

2:  $j \leftarrow 0$                 \Número de conflictos removidos

3:  $L \leftarrow V_f \setminus \{1, n\}$

4: **mientras**  $j < n$  **hacer**

5:    Seleccionar  $k \in L$

6:     $L \leftarrow L \setminus (C_k \cup \{k\})$

7:     $j = j + |C_k| + 1$

8:     $C_k \leftarrow \emptyset$

9:     $C_i \leftarrow \emptyset, \forall i \in C_k$

10: **fin mientras**

**Salida:**  $C_k, k \in V_f \setminus \{1, n\}$  : Conjuntos de conflictos de instancias de clases 4, 5 y 6

---

## B.3 GENERACIÓN DE NODOS OBLIGATORIOS

Para un grafo en particular para el cual se han determinado los conflictos entre nodos, la cardinalidad del conjunto de nodos obligatorios  $M$  está dado por  $m = \lfloor \gamma (|V_f| - d) \rfloor$ , donde  $d$  es el número de depósitos y  $\gamma = 0.1, 0.2$ , y  $0.3$ .

Una vez determinado  $m$ , se selecciona un nodo aleatorio  $k \in V_f$  y si no existe

un nodo en  $C_k \cup \{k\}$  que sea obligatorio, se agrega  $k$  a  $M$ . El proceso continúa hasta que  $|M| = m$ .

Todos los grafos basados en la misma instancia del OP que contienen el mismo porcentaje de nodos obligatorios, también comparten el mismo conjunto  $M$ . El procedimiento anteriormente descrito se aplica a los grafos con nivel de conflictos alto y, posteriormente, se copia el conjunto  $M$  a sus homólogos con nivel de conflicto bajo.

Sea  $I_k$  un grafo perteneciente a la clase de instancias  $k$ , para el cual ya se han fijado los conflictos entre nodos y sea  $V_k$  el conjunto de nodos correspondiente a dicho grafo. El Algoritmo 13 detalla el procedimiento descrito en esta sección.

---

**Algoritmo 13** Fijación de nodos obligatorios
 

---

**Entrada:**

$I_1, I_2, I_3, I_4, I_5, I_6$  : Instancias con conflictos fijados

$d$  : Número de depósitos

1: **para**  $i = 1$  **to**  $i = 6$  **hacer**

2:    $M_i \leftarrow \emptyset$

3: **fin para**

4:  $m_1 = \lfloor 0.1(|V_1| - d) \rfloor$

5:  $m_2 = \lfloor 0.2(|V_2| - d) \rfloor$

6:  $m_3 = \lfloor 0.3(|V_3| - d) \rfloor$

7: **para**  $i = 1$  **to**  $i = 3$  **hacer**

8:   **mientras**  $|M_i| < m_i$  **hacer**

9:     Seleccionar  $k \in V_i$  aleatoriamente

10:    **si**  $(C_k \cup \{k\}) \cap M_i = \emptyset$ , **entonces**

11:      $M_i \leftarrow M_i \cup k$

12:    **fin si**

13:     $M_{i+3} \leftarrow M_i$

14:   **fin mientras**

15: **fin para**

**Salida:**  $M_1, M_2, M_3, M_4, M_5, M_6$ : Conjunto de nodos obligatorios de instancias de clases

---

## B.4 GENERACIÓN DE LA MÁXIMA DURACIÓN DE LA RUTA

Para precisar el valor de la duración máxima del recorrido, se toma como entrada una instancia cuyo conjunto de nodos obligatorios ha sido fijado previamente y se resuelve el TSP sobre dicho conjunto. Sea  $v(TSP')$  el valor de la función así obtenida, entonces el tiempo límite del recorrido estará dado por  $t_{max} = \lambda v(TSP')$ , donde  $\lambda = 1.1, 1.2$  y  $1.3$  para los grafos donde hay un depósito y  $\lambda = 1, 1.1$  y  $1.2$ , en el caso contrario.

Nótese que los grafos correspondientes a las clases 1 y 4, 2 y 5, y 3 y 6 tendrán la misma duración máxima de la ruta, ya que el conjunto de nodos obligatorios es el mismo.

# BIBLIOGRAFÍA

---

- ARCHETTI, C., N. BIANCHESI y M. G. SPERANZA (2013a), «Optimal solutions for routing problems with profits», *Discrete Applied Mathematics*, **161**(4), págs. 547–557.
- ARCHETTI, C., D. FEILLET, A. HERTZ y M. G. SPERANZA (2009a), «The capacitated team orienteering and profitable tour problems», *Journal of the Operational Research Society*, **60**(6), págs. 831–842.
- ARCHETTI, C., D. FEILLET, A. HERTZ y M. G. SPERANZA (2009b), «The capacitated team orienteering and profitable tour problems», *Journal of the Operational Research Society*, **60**(6), págs. 831–842.
- ARCHETTI, C., M. SPERANZA y D. VIGO (2013b), «Vehicle routing problems with profits», *Informe técnico*, Tech. Report WPDEM2013/3, University of Brescia.
- ARKIN, E. M., J. S. MITCHELL y G. NARASIMHAN (1998), «Resource-constrained geometric network optimization», en *Proceedings of the Fourteenth Annual Symposium on Computational Geometry*, ACM, págs. 307–316.
- BALAS, E. (1989), «The prize collecting traveling salesman problem», *Networks*, **19**(6), págs. 621–636.
- BALAS, E. (2007), «The prize collecting traveling salesman problem and its applications», en *The traveling salesman problem and its variations*, Springer, págs. 663–695.

- BEKTAŞ, T. y L. GOUVEIA (2014), «Requiem for the Miller–Tucker–Zemlin sub-tour elimination constraints?», *European Journal of Operational Research*, **236**(3), págs. 820–832.
- BÉRUBÉ, J.-F., M. GENDREAU y J.-Y. POTVIN (2009), «A Branch-and-cut Algorithm for the Undirected Prize Collecting Traveling Salesman Problem», *Networks*, **54**(1), págs. 56–67.
- BOUSSIER, S., D. FEILLET y M. GENDREAU (2007), «An exact algorithm for team orienteering problems», *4OR*, **5**(3), págs. 211–230.
- BUTT, S. E. y T. M. CAVALIER (1994), «A heuristic for the multiple tour maximum collection problem», *Computers & Operations Research*, **21**(1), págs. 101–111.
- BUTT, S. E. y D. M. RYAN (1999), «An optimal solution procedure for the multiple tour maximum collection problem using column generation», *Computers & Operations Research*, **26**(4), págs. 427–441.
- CAMPOS, V., R. MARTÍ, J. SÁNCHEZ-ORO y A. DUARTE (2014), «GRASP with path relinking for the orienteering problem», *Journal of the Operational Research Society*, **65**(12), págs. 1800–1813.
- CHAO, I., B. L. GOLDEN y E. A. WASIL (1996a), «A fast and effective heuristic for the orienteering problem», *European Journal of Operational Research*, **88**(3), págs. 475–489.
- CHAO, I.-M., B. L. GOLDEN y E. A. WASIL (1996b), «The team orienteering problem», *European Journal of Operational Research*, **88**(3), págs. 464–474.
- CHWATAL, A. M. y S. PIRKWIESER (2012), «Solving the Two-Dimensional Bin-Packing Problem with Variable Bin Sizes by Greedy Randomized Adaptive Search Procedures and Variable Neighborhood Search», en R. Moreno-Díaz, F. Pichler y A. Quesada-Arencibia (editores), *Computer Aided Systems Theory EUROCAST 2011, Lecture Notes in Computer Science*, tomo 6927, Springer Berlin Heidelberg, págs. 456–463.

- CLAUS, A. (1984), «A new formulation for the travelling salesman problem», *SIAM Journal on Algebraic Discrete Methods*, **5**(1), págs. 21–25.
- DANTZIG, G., R. FULKERSON y S. JOHNSON (1954), «Solution of a large-scale traveling-salesman problem», *Journal of the operations research society of America*, **2**(4), págs. 393–410.
- DELL'AMICO, M., F. MAFFIOLI y P. VÄRBRAND (1995), «On Prize-collecting Tours and the Asymmetric Travelling Salesman Problem», *International Transactions in Operational Research*, **2**(3), págs. 297–308.
- DHOUB, E., J. TEGHEM, D. TUYTTENS y T. LOUKIL (2013), *A Comparison of Local Search Metaheuristics for a Hierarchical Flow Shop Optimization Problem with Time Lags*, John Wiley & Sons, Inc., págs. 69–96.
- DUARTE, A., J. SÁNCHEZ-ORO, M. G. RESENDE, F. GLOVER y R. MARTÍ (2015), «Greedy randomized adaptive search procedure with exterior path relinking for differential dispersion minimization», *Information Sciences*, **296**(0), págs. 46 – 60.
- FEILLET, D., P. DEJAX y M. GENDREAU (2005), «Traveling salesman problems with profits», *Transportation science*, **39**(2), págs. 188–205.
- FEILLET, D., P. DEJAX, M. GENDREAU y C. GUEGUEN (2004), «An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems», *Networks*, **44**(3), págs. 216–229.
- FEO, T. y M. RESENDE (1995), «Greedy Randomized Adaptive Search Procedures», *Journal of Global Optimization*, **6**(2), págs. 109–133.
- FEO, T. A. y M. G. RESENDE (1989), «A probabilistic heuristic for a computationally difficult set covering problem», *Operations Research Letters*, **8**(2), págs. 67–71.
- FESTA, P. y M. G. RESENDE (2009), «Hybrid GRASP Heuristics», en A. Abraham, A.-E. Hassanien, P. Siarry y A. Engelbrecht (editores), *Foundations of Compu-*

- tational Intelligence Volume 3, Studies in Computational Intelligence*, tomo 203, Springer Berlin Heidelberg, págs. 75–100.
- FINKE, G., CLAUS y E. GUNN (), «A two-commodity network flow approach to the traveling salesman problem», *Combinatorics, Graph Theory and Computing, Proc. 14th South Eastern Conf.*
- FISCHETTI, M., J. J. SALAZAR-GONZALEZ y P. TOTH (1998), «Solving the orienteering problem through branch-and-cut», *INFORMS Journal on Computing*, **10**(2), págs. 133–148.
- FLEURENT, C. y F. GLOVER (1999), «Improved constructive multistart strategies for the quadratic assignment problem using adaptive memory», *INFORMS Journal on Computing*, **11**(2), págs. 198–204.
- FOMIN, F. V. y A. LINGAS (2002), «Approximation algorithms for time-dependent orienteering», *Information Processing Letters*, **83**(2), págs. 57–62.
- FOX, K. R., B. GAVISH y S. C. GRAVES (1980), «Technical Note-An n-Constraint Formulation of the (Time-Dependent) Traveling Salesman Problem», *Operations Research*, **28**(4), págs. 1018–1021.
- GARCIA, A., O. ARBELAITZ, P. VANSTEENWEGEN, W. SOUFFRIAU y M. LINAZA (2010), «Hybrid Approach for the Public Transportation Time Dependent Orienteering Problem with Time Windows», en *Hybrid Artificial Intelligence Systems, Lecture Notes in Computer Science*, tomo 6077, Springer Berlin Heidelberg, págs. 151–158.
- GAVISH, B. y S. C. GRAVES (1978), «The travelling salesman problem and related problems», .
- GENDREAU, M., G. LAPORTE y F. SEMET (1998a), «A branch-and-cut algorithm for the undirected selective traveling salesman problem», *Networks*, **32**(4), págs. 263–273.

- GENDREAU, M., G. LAPORTE y F. SEMET (1998b), «A tabu search heuristic for the undirected selective travelling salesman problem», *European Journal of Operational Research*, **106**(2), págs. 539–545.
- GOLDEN, B. L., L. LEVY y R. VOHRA (1987), «The orienteering problem», *Naval Research Logistics*, **34**(3), págs. 307–318.
- GOUVEIA, L. y J. M. PIRES (1999), «The asymmetric travelling salesman problem and a reformulation of the Miller–Tucker–Zemlin constraints», *European Journal of Operational Research*, **112**(1), págs. 134–146.
- HANSEN, P. y N. MLADENVIĆ (2003), «A tutorial on variable neighborhood search», .
- HAYES, M. y J. NORMAN (1984), «Dynamic programming in orienteering: route choice and the siting of controls», *Journal of the Operational Research Society*, págs. 791–796.
- INTERNATIONAL ORIENTEERING FEDERATION (), «<http://orienteering.org/>», .
- JARBOUI, B., H. DERBEL, S. HANAFI y N. MLADENVIĆ (2013), «Variable neighborhood search for location routing», *Computers & Operations Research*, **40**(1), págs. 47 – 57.
- KANTOR, M. G. y M. B. ROSENWEIN (1992), «The orienteering problem with time windows», *Journal of the Operational Research Society*, págs. 629–635.
- KARBOWSKA-CHILINSKA, J. y P. ZABIELSKI (2013), «A Genetic Algorithm vs. Local Search Methods for Solving the Orienteering Problem in Large Networks», en M. Graña, C. Toro, R. Howlett y L. Jain (editores), *Knowledge Engineering, Machine Learning and Lattice Computing with Applications, Lecture Notes in Computer Science*, tomo 7828, Springer Berlin Heidelberg, págs. 11–20.

- KATAOKA, S. y S. MORITO (1988), «An algorithm for single constraint maximum collection problem», *Journal of the Operations Research Society of Japan*, **31**(4), págs. 515–530.
- KEMMOE TCHOMTE, S. y N. TCHERNEV (2014), «A GRASP<sub>x</sub>ELS for Scheduling of Job-Shop Like Manufacturing Systems and CO<sub>2</sub> Emission Reduction», en B. Grabot, B. Vallespir, S. Gomes, A. Bouras y D. Kiritsis (editores), *Advances in Production Management Systems. Innovative and Knowledge-Based Production Management in a Global-Local World, IFIP Advances in Information and Communication Technology*, tomo 439, Springer Berlin Heidelberg, págs. 130–137.
- KIRLIK, G. y C. OGUZ (2012), «A variable neighborhood search for minimizing total weighted tardiness with sequence dependent setup times on a single machine», *Computers & Operations Research*, **39**(7), págs. 1506 – 1520.
- KUO, Y. y C.-C. WANG (2012), «A variable neighborhood search for the multi-depot vehicle routing problem with loading cost», *Expert Systems with Applications*, **39**(8), págs. 6949 – 6954.
- LABADIE, N., R. MANSINI, J. MELECHOVSKY y R. W. CALVO (2012), «The Team Orienteering Problem with Time Windows: An LP-based Granular Variable Neighborhood Search», *European Journal of Operational Research*, **220**(1), págs. 15 – 27.
- LANGEVIN, A., F. SOUMIS y J. DESROSIERS (1990), «Classification of travelling salesman problem formulations», *Operations Research Letters*, **9**(2), págs. 127–132.
- LAPORTE, G. y S. MARTELLO (1990), «The selective travelling salesman problem», *Discrete Applied Mathematics*, **26**(2), págs. 193–207.
- LEI, H. y K. QIN (2014), «Greedy randomized adaptive search procedure for analog test point selection», *Analog Integrated Circuits and Signal Processing*, **79**(2), págs. 371–383.

- LEIFER, A. C. y M. B. ROSENWEIN (1994), «Strong linear programming relaxations for the orienteering problem», *European Journal of Operational Research*, **73**(3), págs. 517–523.
- LEITNER, M. y G. R. RAIDL (2012), «Variable Neighborhood and Greedy Randomized Adaptive Search for Capacitated Connected Facility Location», en R. Moreno-Díaz, F. Pichler y A. Quesada-Arencibia (editores), *Computer Aided Systems Theory EUROCAST 2011, Lecture Notes in Computer Science*, tomo 6927, Springer Berlin Heidelberg, págs. 295–302.
- LIANG, Y.-C., S. KULTUREL-KONAK y A. E. SMITH (2002), «Meta heuristics for the orienteering problem», en *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, tomo 1, IEEE, págs. 384–389.
- MANERBA, D. y R. MANSINI (2013), «A branch-and-cut algorithm for the Multi-Vehicle Traveling Purchaser Problem with Exclusionary Side Constraints», .
- MARTÍ, R., V. CAMPOS, M. G. RESENDE y A. DUARTE (2015), «Multiobjective GRASP with Path Relinking», *European Journal of Operational Research*, **240**(1), págs. 54 – 71.
- MILLER, C. E., A. W. TUCKER y R. A. ZEMLIN (1960), «Integer programming formulation of traveling salesman problems», *Journal of the ACM (JACM)*, **7**(4), págs. 326–329.
- MLADENOVIĆ, N. y P. HANSEN (1997), «Variable neighborhood search», *Computers & Operations Research*, **24**(11), págs. 1097–1100.
- MLADENOVIĆ, N., D. UROŠEVIĆ, S. HANAFI y A. ILIĆ (2012), «A general variable neighborhood search for the one-commodity pickup-and-delivery travelling salesman problem», *European Journal of Operational Research*, **220**(1), págs. 270 – 285.
- OLIVEIRA, T., V. COELHO, M. SOUZA, D. BOAVA, F. BOAVA, I. COELHO y B. COELHO (2015), «A hybrid variable neighborhood search algorithm for targeted

- offers in direct marketing», *Electronic Notes in Discrete Mathematics*, **47**(0), págs. 205 – 212.
- ORMAN, A. y H. P. WILLIAMS (2006), «A survey of different integer programming formulations of the travelling salesman problem», *Optimisation, Economics and Financial Analysis. Advances in Computational Management Science*, **9**, págs. 93–106.
- PEKNY, J., D. MILLER y G. J. MCRAE (1990), «An exact parallel algorithm for scheduling when production costs depend on consecutive system states», *Computers & Chemical Engineering*, **14**(9), págs. 1009–1023.
- RAMESH, R. y K. M. BROWN (1991), «An efficient four-phase heuristic for the generalized orienteering problem», *Computers & Operations Research*, **18**(2), págs. 151–165.
- RAMESH, R., Y. YOON y K. MH (), «An optimal algorithm for the orienteering problem», .
- RESENDE, M. y J. GONZÁLEZ VELARDE (2003), «GRASP: Greedy randomized adaptive search procedures», *Revista Iberoamericana de Inteligencia Artificial*, **19**.
- RIGHINI, G. y M. SALANI (2006a), «Dynamic programming for the orienteering problem with time windows», *Informe técnico*, Technical report 91, Dipartimento di Tecnologie dell' Informazione, Università degli Studi Milano, Crema, Italy.
- RIGHINI, G. y M. SALANI (2006b), «Symmetry helps: bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints», *Discrete Optimization*, **3**(3), págs. 255–273.
- RIGHINI, G. y M. SALANI (2008), «New dynamic programming algorithms for the resource constrained elementary shortest path problem», *Networks*, **51**(3), págs. 155–170.

- RIGHINI, G. y M. SALANI (2009), «Decremental state space relaxation strategies and initialization heuristics for solving the orienteering problem with time windows with dynamic programming», *Computers & operations research*, **36**(4), págs. 1191–1203.
- SALAZAR-AGUILAR, M. A., A. LANGEVIN y G. LAPORTE (2014), «The multi-district team orienteering problem», *Computers & Operations Research*, **41**, págs. 76–82.
- SCHILDE, M., K. F. DOERNER, R. F. HARTL y G. KIECHLE (2009), «Metaheuristics for the bi-objective orienteering problem», *Swarm Intelligence*, **3**(3), págs. 179–201.
- SEVKLI, Z. y F. SEVILGEN (2006), «Variable Neighborhood Search for the Orienteering Problem», en A. Levi, E. Sava, H. Yenigün, S. Balcisoy y Y. Saygin (editores), *Computer and Information Sciences - ISCIS 2006, Lecture Notes in Computer Science*, tomo 4263, Springer Berlin Heidelberg, págs. 134–143.
- SEVKLI, Z. y F. E. SEVILGEN (2010), «Discrete particle swarm optimization for the orienteering Problem», en *Evolutionary Computation (CEC), 2010 IEEE Congress on*, IEEE, págs. 1–8.
- SILVA, R. M., M. G. RESENDE, P. FESTA, F. L. VALENTIM y F. N. JUNIOR (2014), «GRASP with Path-Relinking for the Maximum Contact Map Overlap Problem», en P. M. Pardalos, M. G. Resende, C. Vogiatzis y J. L. Walteros (editores), *Learning and Intelligent Optimization, Lecture Notes in Computer Science*, tomo 8426, Springer International Publishing, págs. 223–226.
- SMITH, D. E. (2004), «Choosing Objectives in Over-Subscription Planning.», en *ICAPS*, tomo 4, pág. 393.
- SOUFFRIAU, W. y P. VANSTEENWEGEN (2010), «Tourist Trip Planning Functionalities: State-of-the-Art and Future», en F. Daniel y F. Facca (editores), *Current*

- Trends in Web Engineering, Lecture Notes in Computer Science*, tomo 6385, Springer Berlin Heidelberg, págs. 474–485.
- SOUFFRIAU, W., P. VANSTEENWEGEN, J. VERTOMMEN, G. V. BERGHE y D. V. OUDHEUSDEN (2008), «A personalized tourist trip design algorithm for mobile tourist guides», *Applied Artificial Intelligence*, **22**(10), págs. 964–985.
- SUN, M. (2002), «The transportation problem with exclusionary side constraints and two branch-and-bound algorithms», *European Journal of Operational Research*, **140**(3), págs. 629 – 647.
- TANG, H. y E. MILLER-HOOKS (2005), «A TABU search heuristic for the team orienteering problem», *Computers & Operations Research*, **32**(6), págs. 1379 – 1407.
- TASGETIREN, M. y A. E. SMITH (2000), «A genetic algorithm for the orienteering problem», en *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, tomo 2, IEEE, págs. 910–915.
- TRICOIRE, F., M. ROMAUCH, K. F. DOERNER y R. F. HARTL (2010), «Heuristics for the multi-period orienteering problem with multiple time windows», *Computers & Operations Research*, **37**(2), págs. 351–367.
- TSILIGIRIDES, T. (1984), «Heuristic methods applied to orienteering», *Journal of the Operational Research Society*, **35**(9), págs. 797–809.
- VAJDA, S. (1961), «MATHEMATICAL PROGRAMMING..», .
- VANSTEENWEGEN, P. (2009), «Planning in tourism and public transportation», *4OR*, **7**(3), págs. 293–296.
- VANSTEENWEGEN, P., W. SOUFFRIAU, G. V. BERGHE y D. V. OUDHEUSDEN (2009), «Iterated local search for the team orienteering problem with time windows», *Computers & Operations Research*, **36**(12), págs. 3281 – 3290.

- VANSTEENWEGEN, P., W. SOUFFRIAU, G. V. BERGHE y D. V. OUDHEUSDEN (2011a), «The City Trip Planner: An expert system for tourists», *Expert Systems with Applications*, **38**(6), págs. 6540 – 6546.
- VANSTEENWEGEN, P., W. SOUFFRIAU y D. V. OUDHEUSDEN (2011b), «The orienteering problem: A survey», *European Journal of Operational Research*, **209**(1), págs. 1–10.
- WANG, Q., X. SUN, B. L. GOLDEN y J. JIA (1995), «Using artificial neural networks to solve the orienteering problem», *Annals of Operations Research*, **61**(1), págs. 111–120.
- WESTERLUND, A., M. GÖTHE-LUNDGREN y T. LARSSON (2006), «A stabilized column generation scheme for the traveling salesman subtour problem», *Discrete Applied Mathematics*, **154**(15), págs. 2212–2238.
- WONG, R. T. (1980), «Integer Programming Formulations of the Traveling Salesman Problem», *Proceedings of the IEEE International Conference on Circuits and Computers*, págs. 149–152.

# FICHA AUTOBIOGRÁFICA

---

Pamela Jocelyn Palomo Martínez

Candidato para el grado de Maestra en Ciencias  
en Ingeniería de Sistemas

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Tesis:

## PROBLEMA DEL AGENTE VIAJERO SELECTIVO CON RESTRICCIONES ADICIONALES

Nací el 10 de febrero de 1990 en Ciudad Madero, Tamaulipas, primogénita de María Concepción Martínez Guerrero y Astolfo Palomo Solís.

En 1996 comencé mi educación primaria en la Escuela Primaria Benito Juárez en Pánuco, Veracruz, la cual concluí en el 2002 en la Escuela Primaria del Estado Ford 74 en Ciudad Victoria, Tamaulipas. Inicié mi educación secundaria en la Escuela Secundaria General No. 4 “Profesor José Santos Valdés Salazar”. Concluí mi educación secundaria en el año 2005 y fui merecedora de la medalla al mérito “José Santos Valdés Salazar”. Posteriormente decidí estudiar un bachillerato físico matemático en la carrera técnica en electrónica en el Centro de Bachillerato Tecnológico industrial y de servicios No. 24 en Ciudad Victoria, Tamaulipas. En el año 2008 me titulé como técnica, siendo merecedora de la medalla al mérito en matemáticas “Elvia Vázquez

Flores” y del reconocimiento al mejor estudiante de la carrera técnica en electrónica.

En el año 2008 ingresé a la Facultad de Ciencias Físico Matemáticas de la Universidad Autónoma de Nuevo León. El año 2011 cursé la materia de investigación de operaciones, la cual me atrajo a tal grado de que en ese mismo año fui prestamista de servicio social en el programa “Uso de algoritmos para resolver problemas de programación” en el Centro de Investigación en Ciencias Físico Matemáticas de la Facultad de Ciencias Físico Matemáticas de la Universidad Autónoma de Nuevo León, bajo la dirección del Dr. Fernando Camacho.

Posterior a la terminación de mi servicio social, seguí trabajando con el Dr. Camacho en mi tesis de licenciatura la cual consistió en la propuesta de una metaheurística Stackelberg-Evolutiva para resolver el problema binivel de fijación de cuotas en una red de transporte. Gracias a este trabajo, tuve la oportunidad de participar en dos congresos nacionales en las ciudades de Guadalajara y Santiago de Querétaro.

En el año 2012 concluí mis estudios de licenciatura e ingresé como estudiante de maestría al Programa de Posgrado en Ingeniería de Sistemas de la Facultad de Ingeniería Mecánica y Eléctrica de la Universidad Autónoma de Nuevo León en el año 2013.

Durante mis estudios de posgrado he tenido la oportunidad de participar en congresos nacionales e internacionales en las ciudades de Monterrey, Guanajuato, Mérida y Valparaiso; así como de impartir cursos y talleres en la facultad, de asistir a una escuela de matemáticas discretas en Guanajuato y de realizar una estancia de investigación en la Universidad Técnica Federico Santa María en Santiago de Chile.