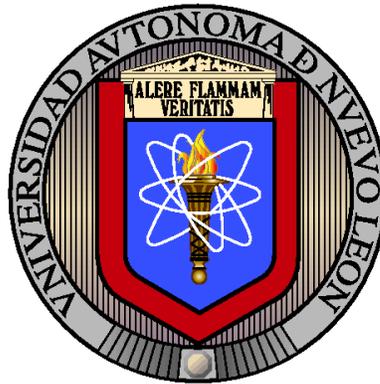


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

DIVISIÓN DE ESTUDIOS DE POSGRADO



COTAS LAGRANGIANAS PARA EL
PROBLEMA DE RUTEO DE VEHÍCULOS
EN RED TIPO ESTRELLA

POR

LIC. LUIS ALFONSO INFANTE RIVERA

EN OPCIÓN AL GRADO DE

MAESTRÍA EN CIENCIAS

EN INGENIERÍA DE SISTEMAS

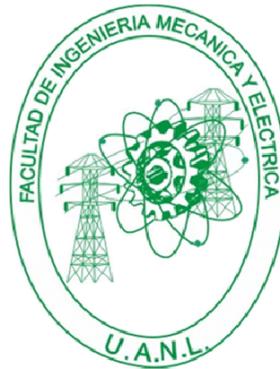
SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN

DICIEMBRE 2012

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

DIVISIÓN DE ESTUDIOS DE POSGRADO



COTAS LAGRANGIANAS PARA EL
PROBLEMA DE RUTEO DE VEHÍCULOS
EN RED TIPO ESTRELLA

POR

LIC. LUIS ALFONSO INFANTE RIVERA

EN OPCIÓN AL GRADO DE

MAESTRÍA EN CIENCIAS

EN INGENIERÍA DE SISTEMAS

SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN

DICIEMBRE 2012

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

Los miembros del Comité de Tesis recomendamos que la Tesis «COTAS LAGRANGIANAS PARA EL PROBLEMA DE RUTEO DE VEHÍCULOS EN RED TIPO ESTRELLA», realizada por el alumno Lic. Luis Alfonso Infante Rivera, con número de matrícula 0849084, sea aceptada para su defensa como opción al grado de Maestría en Ciencias en Ingeniería de Sistemas.

El Comité de Tesis

Dr. Igor S. Litvinchev

Asesor

Dr. Oscar Leonel Chacon Mondragon

Revisor

Dr. Jose Antonio Marmolejo Saucedo

Revisor

Vo. Bo.

Dr. Moises Hinojosa Rivera

División de Estudios de Posgrado

San Nicolás de los Garza, Nuevo León, diciembre 2012

*Aquí puedes poner tu dedicatoria
si es que tienes una.*

*Si no tienes una, puedes borrar
la línea `\include{Dedicatoria}` en el
archivo `MiTesis.tex` pues no es obligatoria.*

ÍNDICE GENERAL

Agradecimientos	XIII
Resumen	XIV
1. Introducción	1
1.1. Estructura de la tesis	1
1.2. Problema de ruteo de vehículos (VRP)	2
1.2.1. Problema de ruteo de vehículos clásico y otros	2
1.2.2. Problema de ruteo de vehículos tipo estrella (VRP-Starcase)	4
1.3. Objetivo	6
1.4. Justificación	6
1.5. Hipótesis	7
1.6. Metodología	7
2. Revisión de la literatura	8
2.1. Relaxaciones a problemas de optimización	8
2.1.1. Relaxación lineal	9
2.1.2. Relaxación subrogada	9

2.2. Relajación Lagrangiana	10
2.2.1. La función lagrangiana	12
2.3. Propiedades de la Relajación Lagrangiana	13
2.3.1. solución factible lagrangeana	13
2.3.2. Propiedad de Integralidad	13
2.3.3. descomposición en subproblemas	14
2.4. Descomposición lagrangiana	14
2.5. Relajación Lagrangiana modificada	16
2.6. métodos para resolver la Relajación Lagrangiana	17
3. Estado del arte	25
3.1. Introducción	25
3.2. Problema de VRP con ventanas de tiempo suaves	25
3.3. Problema de VRP con entregas divididas	26
3.4. Problema de ruteo en minas	26
3.5. Problema de ruteo de contenedores	28
3.6. Abastecimiento de comida en aviones	28
3.7. Problema de ruteo de vehículos de emergencias medicas	30
3.8. Maquinas paralelas	30
4. Formulación matemática	32
4.1. Introducción	32
4.2. Modelo matemático	32

4.2.1. función Objetivo	35
4.2.2. Restricciones	35
5. Construcción de cotas	40
5.1. Introducción	40
5.2. relajación 1er grupo de restricciones	40
5.3. relajación 2do grupo de restricciones	41
5.4. relajación 3er grupo de restricciones	42
5.5. relajación 4to grupo de restricciones	43
5.6. relajación 5to y 6to grupo de restricciones	43
5.7. Método del subgradiente - detallado	44
5.8. Sobre el GAP	47
6. Implementación computacional	51
6.1. Introducción	51
6.2. GAMS	51
6.3. Ilog Cplex Optimization Studio	51
6.4. Librería Cplex	52
6.5. Lenguaje C++	54
6.6. archivos de datos	54
6.7. archivos de salida	55
6.8. Algoritmos	55
6.9. Simulación	60

7. Experimentación computacional	62
7.1. Introducción	62
7.2. Creación de instancias	62
7.3. Experimentacion	64
7.3.1. Parámetros del subgradiente	64
7.3.2. Calculo del GAP	65
7.4. Resumen resultados	65
7.5. Sobre la cota lagrangeana	68
7.5.1. Resumen de resultados	69
8. Conclusiones	72
8.1. Trabajo futuro	73
9. TABLAS	74
9.1. Soluciones Cplex	74
9.2. Parametros del subgradiente	76
9.3. Soluciones - Relajacion en grupo G0	78
9.4. Soluciones - Relajación en grupo G1 subgradiente normal	80
9.5. Soluciones - Relajación en grupo G1 subgradiente modificado	82

ÍNDICE DE FIGURAS

1.1. Problema de ruteo de vehículos como un grafo completo	3
1.2. Problema de ruteo de vehículos tipo Star-case, grafo incompleto . . .	6
2.1. Interpretación geométrica de la Relajación Lagrangiana	13
2.2. Interpretación geométrica de la descomposición lagrangiana	15
2.3. Ilustración del método del gradiente	18
3.1. Ruteo en minas, evitar conflictos entre vehículos	27
3.2. Problema de ruteo de contenedores	29
4.1. Descripción de función objetivo	35
4.2. Descripción de restricción 1	36
4.3. Descripción de restricción 2	36
4.4. Descripción de restricción 3	37
4.5. Descripción de restricción 4	37
4.6. Descripción de restricción 5	38
4.7. Descripción de restricción 6	39

5.1. GAP entre solución factible y la óptima global	48
5.2. Descripción del GAP relativo a una solución lineal	49
5.3. Descripción del GAP relativo a una solución lagrangiana	50

ÍNDICE DE TABLAS

7.1. Tipos de instancias	63
7.2. GAPs de soluciones Cplex para 8 horas y 1 hora	64
7.3. GAPs Tipo A	66
7.4. GAPs Tipo B	66
7.5. GAPs Tipo C	67
7.6. GAPs Tipo D parte 1	67
7.7. GAPs Tipo D parte 2	68
7.8.	69
7.9.	69
7.10.	70
7.11.	70
7.12.	71
9.1. Tabla Soluciones Cplex	74
9.2. Tabla Soluciones Cplex - continuacion	75
9.3. Parametros subgradiente G1	76

9.4. Parametros subgradiente G0	77
9.5. Relajaciones G0	78
9.6. Relajaciones G0 continuacion	79
9.7. Relajaciones G1	80
9.8. Relajaciones G1	81
9.9. Relajaciones G1	82
9.10. Relajaciones G1	83

AGRADECIMIENTOS

Aquí puedes poner tus agradecimientos. (No olvides agradecer a CONACyT en caso de que hallas sido beneficiado con una beca).

RESUMEN

Lic. Luis Alfonso Infante Rivera.

Candidato para el grado de Doctor en Ingeniería
con especialidad en Ingeniería de Sistemas.

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Título del estudio:

COTAS LAGRANGIANAS PARA EL PROBLEMA DE RUTEO DE VEHÍCULOS EN RED TIPO ESTRELLA

Número de páginas: 88.

OBJETIVOS Y MÉTODO DE ESTUDIO: Aquí debes poner tus objetivos y métodos de estudio. (Este es el formato).

CONTRIBUCIONES Y CONCLUSIONES: Y aquí tus contribuciones y conclusiones. (También es parte del formato).

Firma del asesor: _____

Dr. Igor S. Litvinchev

CAPÍTULO 1

INTRODUCCIÓN

1.1 ESTRUCTURA DE LA TESIS

- En el primer capítulo se presenta el objetivo de la tesis, las hipótesis implicadas, la justificación de la tesis y la metodología empleada para el desarrollo de la misma, además de una breve introducción al problema de ruteo en general y al caso de esta tesis.
- En el segundo capítulo está dedicado a la explicación de conceptos como relajación y en específico la relajación lagrangiana, su utilidad y metodología para resolver los problemas que implica (la resolución del dual lagrangiano)
- En el tercer capítulo se discutirá sobre el estado actual del arte sobre los problemas de ruteo o los que están relacionados con nuestro modelo de investigación
- En el cuarto capítulo se explica el modelo matemático del problema de ruteo de vehículo tipo estrella.
- En el quinto capítulo se explica detalladamente la manera en que se construyeron los modelos relajados para el cálculo de cotas lagrangianas, normales y modificadas, además de la estimación del error relativo GAP que obtendremos con esta cota.
- Para el sexto capítulo se hablará sobre la implementación computacional realizada para llevar a cabo las relajaciones.

- En el séptimo capítulo incluye los experimentos computacionales y algunas estimaciones con los resultados de estos.
- En el octavo capítulo se presentan las conclusiones obtenidas de esta investigación y algunas sugerencias para trabajo futuro.
- El noveno capítulo presenta más detalladamente los datos de las instancias y los parámetros utilizados en las relajaciones.

1.2 PROBLEMA DE RUTEO DE VEHÍCULOS (VRP)

El antecedente histórico del problema de ruteo de vehículos data de finales de la década de los 50's del siglo XX; Dantzing y Ramser en 1959 [11] establecieron una formulación matemática como una generalización del Problema del Agente Viajero (TSP) presentado por Flood[15], y un algoritmo de resolución para dar una solución cercana a la optima para el problema de suministro de gasolina en las estaciones de servicio, en el año 1964 Clarke y Wriqth propusieron una heurística voraz (conocida como el algoritmo de los ahorros [9]), desde entonces cientos de algoritmos, modelos han sido propuestos para solucionar este problema y sus diferentes versiones por este tipo de problemas envuelve a los investigadores hasta el día de hoy [39]. En el problema de ruteo de vehículos se establece que desde un depósito (o centro de suministros) parten vehículos cargados con mercancías para ser llevadas a diversos clientes y así satisfacer la demanda de los mismos, el problema es desarrollar las rutas óptimas que tengan que hacer los vehículos para satisfacer la demanda de los clientes

1.2.1 PROBLEMA DE RUTEO DE VEHÍCULOS CLÁSICO Y OTROS

En muchas organizaciones el manejo de la distribución de las actividades constituye un problema de decisión mayor; la utilización eficiente de una flotilla de vehículo, del ruteo de estos, es la parte más importante en los problemas de ruteo. Preguntas

como cuantos y de que capacidad de vehículos se necesitan para obtener por ejemplo un costo mínimo es lo que necesita responderse. El problema de ruteo de vehículos clásico, presentado en la sección anterior, tiene como restricciones clásicas, el que un vehículo en su viaje hacia los clientes (porque puede visitar varios en el mismo viaje), El problema de ruteo de vehículos puede ser reducido a hallar el modo en que la distancia recorrida por los vehículos sea la menor posible y usar el menor número de estos, en la interpretación clásica del VRP se permite conexiones entre clientes, proporcionando un grafo completo, al problema de una red incompleta o cuando podría no haber conexiones entre todos los nodos o clientes, se le ha puesto poca atención, ver figura 1.1.

VRP Clasico

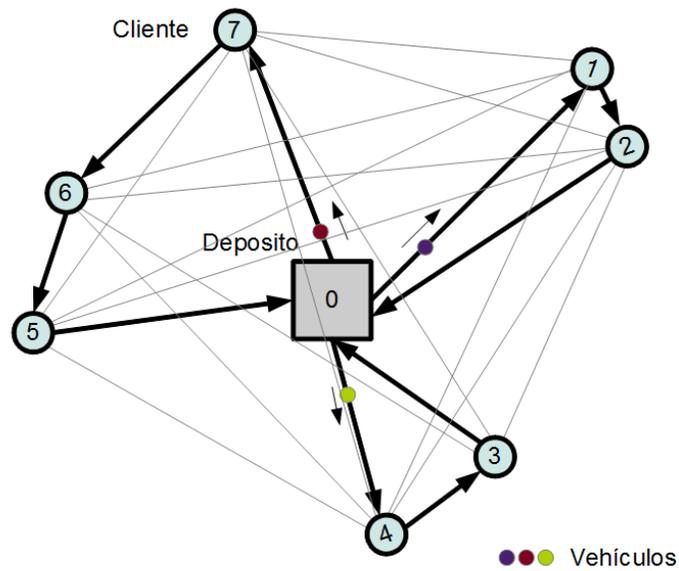


Figura 1.1: Problema de ruteo de vehículos como un grafo completo

Sin embargo en la vida real el problema de ruteo de vehículos clásico es mucho más complejo, en la práctica se agregan quitan o modifican restricciones complicando o simplificando su resolución.

Una de estas variantes es el CVRP (Capacitated Vehicle Routing Problem) en

este problema los vehículos tienen una capacidad máxima de mercancía, esta variante es considerada la más conocida y estudiada, la formulación matemática de Dantzing y Ramer es para un problema de CVRP [11], otra variante es el VRPTW (Vehicle Routing Problem with Time Windows), la cual impone una restricción de tiempo en la llegada de vehículos a los clientes, lo que hace que el problema de VRPTW también tenga un problema de Scheduling (o secuenciación) por lo que también es conocido como VRPSTW, el primer caso de estudio fue presentado en 1967 por Pullen y Webb [31] y una variante adicional es la combinación de ambos la cual es llamada CVRPTW (Capacitated Vehicle Routing Problem with Time Windows)[40].

El VRP es un problema tipo NP-Completo (Lenstra y Rinnooy Kan, 1981 [27]), y solo puede ser resuelto en un tiempo razonable en instancias pequeñas, las heurísticas no garantizan una solución exacta aunque dan buenos resultados en la práctica, En los últimos 20 años se han desarrollado de forma prometedora muchas metaheurísticas para la solución del problema, el libro de Golden del 2008 [5] sobre los últimos avances y técnicas para la resolución del problema de VRP se pueden encontrar algunas de estas.

El uso de métodos aproximados para la resolución de este tipo de problemas plantea la necesidad de tener un medio para evaluar las soluciones dadas; las cotas inferiores y superiores son útiles para la determinación de que tan buena es una solución y sustentan la base de este trabajo de investigación.

1.2.2 PROBLEMA DE RUTEO DE VEHÍCULOS TIPO ESTRELLA (VRP-STARCASE)

Para este trabajo de investigación nos hemos enfocado en una variante del problema de ruteo de vehículos llamado VRP-Starcase o problema de ruteo de vehículos en red tipo estrella. En esta variante existe el problema de la capacidad de los vehículos y el problema de la ventana de tiempo de entrega de los clientes, siendo la ventana de tiempo suave que no impide que los clientes lleguen fuera de esta ventana incurriendo en multa si esto ocurre.

Una característica importante en esta variante es que el vehículo puede hacer varios viajes para satisfacer la demanda del cliente y que el vehículo solo puede visitar a un cliente en cada uno de sus viajes.

La función de costo implica optimizar el costo de los viajes, el costo por las multas y el costo por adquirir los vehículos que se vayan realmente a utilizar. Este modelo se da en la práctica por diversas situaciones; por ejemplo, algunas de éstas son:

- La red de transporte tiene una configuración tipo estrella, esta situación ocurre en redes bajo tierra, por ejemplo (minas, tren subterráneo, etc.), donde efectivamente no existe una conexión física entre clientes, todas las conexiones pasan a través del centro.
- La red de transporte puede ser un grafo completo, pero por reglas de la empresa no se permite utilizar la conexión entre los clientes, aunque esta exista; por ejemplo en el suministro de gasolina de PEMEX (Empresa de petróleo de México), la demanda de gasolina de una estación es varias veces más grande que la capacidad del vehículo y por razones de seguridad se exige retornar al centro inmediatamente después de visitar una estación de gasolina. En general estas reglas pueden deberse a razones de seguridad (por el material peligroso que se transporta) o también por el valor de la mercancía.
- Otra causa probable para que se dé una red tipo estrella es que la demanda de los clientes sea tan grande, mucho más grande que la capacidad de los vehículos, que haga imposible satisfacer la demanda en una sola entrega; así que, aunque la red tenga un grafo completo, las conexiones cliente a cliente no se utilizan, ya que el vehículo queda vacío después de entregar a un cliente obligando a regresar al depósito para recarga.

Una representación gráfica del problema VRP Star-case se presenta en la figura 1.2.

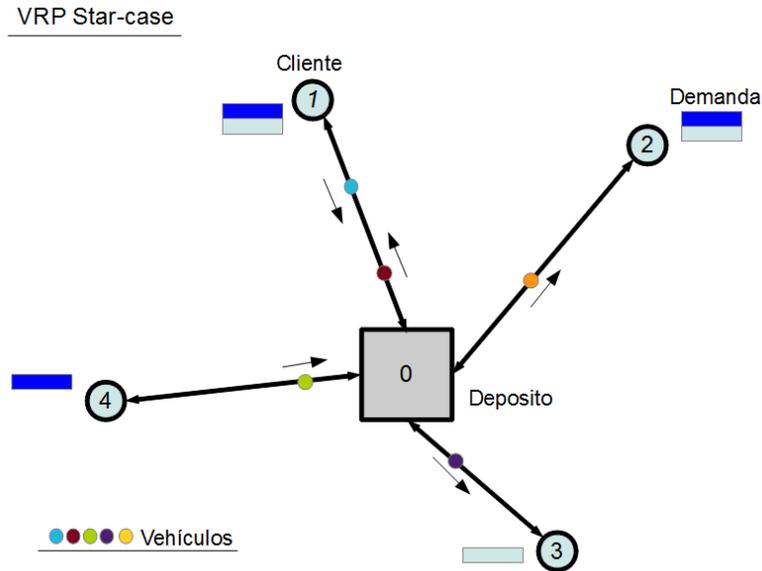


Figura 1.2: Problema de ruteo de vehículos tipo Star-case, grafo incompleto

1.3 OBJETIVO

El objetivo de esta investigación es encontrar mejores cotas (mediante el cálculo de las cotas lagrangeanas del problema) que las ofrecidas por software comercial, para el problema de ruteo de vehículos en red tipo estrella, desarrollando una herramienta de cálculo para el cálculo de estas cotas, para este tipo de problema.

1.4 JUSTIFICACIÓN

La solución de este modelo no es sencilla, consumiendo mucho tiempo en un solver comercial pudiendo llevar días llegar a la solución óptima global. Las soluciones factibles intermedias, obtenidas en un tiempo razonable de ejecución son evaluadas por el solver (cplex) como lejanas a la solución óptima global[28]. La justificación de este trabajo, está en poder ofrecer una mejor estimación de que tan cerca esta una solución factible de la solución óptima global, dada por el solver (o un heurístico).

1.5 HIPÓTESIS

Para el cálculo de las cotas lagrangeanas este trabajo se basa en diversos estudios que se han hecho sobre la relajación lagrangiana; la teoría dada por Guignard [21] y la teoría aplicada a problemas de programación lineal como por ejemplo los trabajos sobre relajaciones lagrangianas en problemas de asignación de Saucedo [29]. La teoría nos marca que podemos obtener una cota mejor que la cota calculada por el solver cplex. La teoría en que se sustenta esta hipótesis es detallada en el segundo capítulo.

1.6 METODOLOGÍA

La metodología que se propone es construir instancias del modelo conforme al perfil de características de casos reales, calcular la mejor solución que se pueda obtener dentro de un límite de tiempo y finalmente su cota lagrangiana (de ser posible) para cada restricción del modelo con el método del subgradiente.

CAPÍTULO 2

REVISIÓN DE LA LITERATURA

En este capítulo se detallará la teoría en que se basa nuestra tesis para el cálculo de cotas de nuestro problema, explicando brevemente que se entiende cuando hablamos de relajaciones y que relajaciones existen (además de la lagrangiana). Se da además una explicación de los métodos utilizados para resolver el problema del dual lagrangiano (o cálculo de cota lagrangiana)

2.1 RELAJACIONES A PROBLEMAS DE OPTIMIZACIÓN

En un problema de programación lineal entera o mixta, la obtención de la solución se comporta de la siguiente manera, al crecer el tamaño del problema, aumenta además el tiempo de cálculo para obtener su solución óptima, pero el tiempo de cálculo para este tipo de problemas, no crece de manera aritmética como se podría pensar, crece de una manera exponencial, haciendo difícil llegar a una solución óptima en un tiempo conveniente, aún en problemas que incluso no parecen muy grandes. Los problemas no solo se complican por su tamaño, sino que su misma estructura interna y la característica discreta de las variables (enteras o binarias) puede hacer complejo un problema. Para la solución de problemas lineales complejos como los anteriores, muchas veces no es posible ofrecer una solución óptima global, de hecho el hallar buenas soluciones puede ser tardado con los métodos matemáticos convencionales (simplex, método de puntos interiores etc.).

Las heurísticas o metaheurísticas dan buenas soluciones, sin embargo nunca se ase-

gura que sean las óptimas globales, se habla de buenas soluciones, de óptimos locales, además las heurísticas y metaheurísticas pueden ser muy específicas para cada tipo de problema[35].

Otro enfoque para dar solución a un problema, es relajarlo; cuando se habla de una relajación, se habla de modificar el modelo matemático del problema, simplificarlo, hacerlo menos restrictivo, ya sea quitando restricciones, volviendo continuas las variables enteras, agrupando restricciones en una sola. Las relajaciones nos ofrecen soluciones modificadas al problema, soluciones que nos pueden servir de cotas para los métodos exactos o como punto de partida para encontrar buenas soluciones factibles combinándolas con algún método heurístico, por si mismas las relajaciones pueden dar solución problema en algunos caso.

2.1.1 RELAJACIÓN LINEAL

Cuando se habla de la relajación lineal de un problema, se habla de convertir las variables enteras de nuestro problema en variables continuas[?]; mediante esa simplificación el tiempo de cálculo de un problema se reduce considerablemente, un problema puede resolverse solo con la utilización del método simplex. Sin embargo la Relajación lineal ofrece una cota muy pobre para los problemas y a pesar de no ofrecer una buena cota o de no aproximar a una buena solución en la mayoría de los casos, se puede utilizar ya sea como referencia a otras cotas o como parte de un método mas grande por ejemplo en el desarrollo del algoritmo de branch and bound o también en el algoritmo de solución en una relajación lagrangiana.

2.1.2 RELAJACIÓN SUBROGADA

Este tipo de Relajación consiste en agrupar varias restricciones en una sola, Glover (1975) [20] desarrolló una teoría de sobre estas mismas ya que no existía un cuerpo teórico completo sobre estas como si lo era para la Relajación lagrangiana. En este tipo de Relajación se trata de hallar como agrupar de la manera mas conve-

niente las restricciones, como seleccionar el mejor agrupamiento, ya sea para hallar la solución al problema o para encontrar buenas cotas.

Si tenemos un modelo matemático definido como

$$(P) \quad \min_x \{fx | Ax \geq 0, x \in X\} \quad (2.1)$$

Donde fx es la función objetivo y $Ax \geq 0$ define un conjunto de restricciones, entonces podemos definir una restricción subrogada como la combinación lineal de todo el conjunto de restricciones, asociamos cada restricción con un multiplicador u_i para cada restricción del conjunto, así que nuestro modelo subrogado sería

$$(SP_u) \quad \min_x \{fx | uAx \geq 0, x \in X\} \quad (2.2)$$

Para poder mantener la desigualdad \geq en la restricción, se fuerza a que todos los multiplicadores sean $u \geq 0$, ya que este modelo es relajado ofrecerá una solución menor a la óptima del modelo primal. De esta forma la solución del problema dual será hallar el conjunto óptimo de multiplicadores u que de la solución máxima del modelo subrogado.

$$(SD) \quad \max_{u \geq 0} \{SP(u)\} \quad (2.3)$$

Relajaciones subrogadas han sido aplicadas en los trabajos de Dinkel y Kochenberger[?] y Chen et al [?]. Se han hecho también combinaciones del planteamiento de la Relajación Subrogada con el de la Relajación Lagrangiana (Espejo y Galvao[?] y Caprara et al[7]).

2.2 RELAJACIÓN LAGRANGIANA

La Relajación lagrangiana en la teoría clásica dada por Guignard[21], Fisher[?], se habla de retirar restricciones del modelo original (el solo prescindir de restricciones ya nos habla de una existencia de un tipo de Relajación). Si además estas restricciones las usamos en la función objetivo como penalización, esta introducción de penalización convierte al problema en una Relajación lagrangiana.

La función objetivo penalizada no solo va a depender de las variables del problema, sino de los llamados multiplicadores lagrangianos y el valor optimizado de esta función dependerá del valor de estos multiplicadores. Una explicación mas formal se da a continuación.

Primeramente definimos nuestro problema al que llamaremos P y sera nuestra definición original o primal del problema, la cual tendrá dos tipos de restricciones, la solución óptima de este problema la llamaremos z^* o también $v(P)$ como se muestra a continuación.

$$(P) \quad \min_x \{fx | Ax \leq b, Cx \leq d, x \in X\} \quad (2.4)$$

Las variables x de nuestro problema son restringidas a pertenecer al conjunto X el cual estará restringido a variables enteras o binarias. Para pasar nuestro modelo a un modelo de forma lagrangeana, bastará con pasar un conjunto de las restricciones a la función objetivo, multiplicando éstas con un vector de multiplicadores λ . La definición de nuestro problema quedara de la siguiente manera.

$$(LR_\lambda) \quad \min_x \{fx + \lambda(Ax \leq b) | Cx \leq d, x \in X\} \quad (2.5)$$

La elección de cual conjunto de restricciones se relajará deberá al conjunto de restricciones difícil que al ser quitado simplificara mucho nuestro problema; en este caso suponemos que $Ax \leq b$ es el conjunto de restricciones difícil, Debido a la naturaleza de las restricciones que son del tipo " $(Ax - b) \leq 0$ ", los multiplicadores λ tendrán que ser positivos $\lambda \geq 0$, el problema modificado ya sin el operador de desigualdad, queda expresada como sigue.

$$(LR_\lambda) \quad \min_x \{fx + \lambda(Ax - b) | Cx \leq d, x \in X\} \quad (2.6)$$

LR_λ es el modelo matemático del problema relajado. La expresión de la función objetivo en este problema LR_λ queda dependiente del valor de los multiplicadores λ . La evaluación de esta función para dado λ dará un valor al que llamaremos Lz que podría ser igual o menor a la solución óptima del problema original, esto es $Lz \leq z^*$. El cálculo de la cota lagrangiana se reduce a la solución del llamado problema dual lagrangiano, que en el caso de minimización no es mas que la búsqueda

del máximo valor de la función objetivo penalizada a minimizar, buscando entre todos los multiplicadores posibles, a este problema de búsqueda se le llama dual lagrangiano y queda expresado de la siguiente forma:

$$\max_{\lambda \geq 0} v(LR_\lambda) \quad (2.7)$$

donde $v(LR_\lambda)$ representa una solución al modelo relajado para dado un multiplicador λ . Al valor resultante de la solución del dual lagrangiano le llamaremos Lz^* y deberá ser igual o mayor a cualquier valor Lz , pero menor o igual al valor óptimo z^* del problema primal, esto es $Lz \leq Lz^* \leq z^*$. La solución del dual lagrangiano es la que nos da la cota lagrangiana que es el problema de esta tesis. Para encontrar esta solución al dual lagrangiano, lo que no es un problema trivial y para resolverlo se utilizan diversos métodos.

Otra forma de expresar el problema del dual lagrangiano es describirlo de una manera equivalente con este modelo primal

$$\min_x \{fx \mid Ax \leq b, x \in Co\{x \in X \mid Cx \leq d\}\} \quad (2.8)$$

Esta forma de expresar el dual lagrangiano, nos dice que la solución estará en la intersección de la envolvente convexa de las restricciones no relajadas y de la envolvente de las restricciones relajadas (ver figura 2.1).

2.2.1 LA FUNCIÓN LAGRANGIANA

En la sección anterior se menciona el hecho de que en la Relajación Lagrangiana un modelo relajado queda en función de sus multiplicadores, esto nos dice que para dado un conjunto de multiplicadores existe una solución óptima del modelo relajado. La solución Lz queda entonces en función de los multiplicadores $Lz(\lambda) = v(LR_\lambda)$ y la solución del dual lagrangiano sería $Lz^* = \max_{\lambda \geq 0} v(LR_\lambda)$. El objetivo de esta función, es encontrar en el caso de minimización, el máximo de los mínimos calculados del modelo relajado. en una representación gráfica de las soluciones del modelo relajado (λ vs Lz) sería el punto máximo del gráfico (caso minimización).

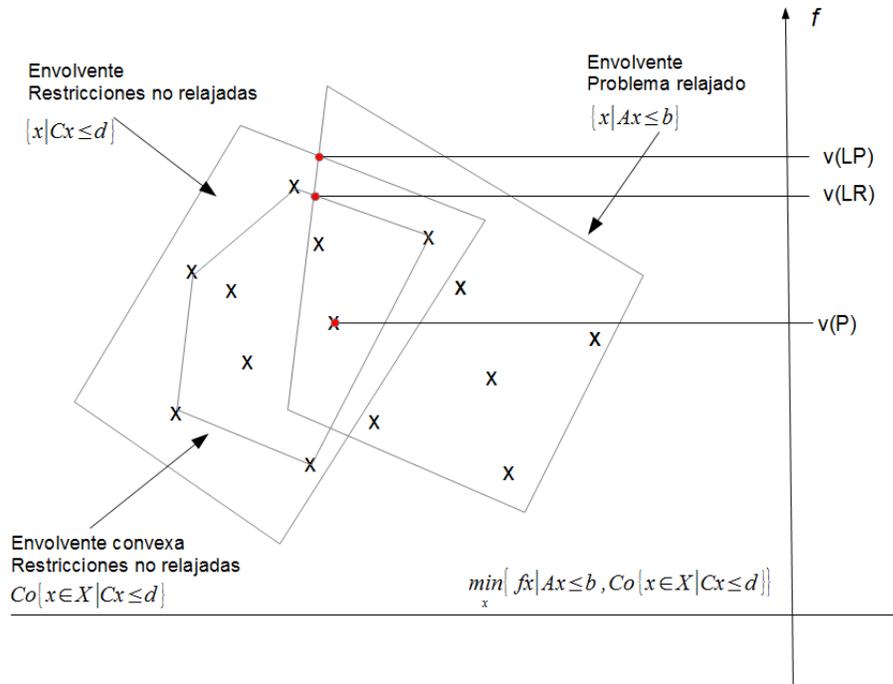


Figura 2.1: Interpretación geométrica de la Relajación Lagrangiana

2.3 PROPIEDADES DE LA RELAJACIÓN LAGRANGIANA

2.3.1 SOLUCIÓN FACTIBLE LAGRANGIANA

Una de las propiedades interesantes del problema dual lagrangiano son que si la solución de esta es factible y además el término de holgura complementaria es $\lambda(Ax - b) = 0$, esta solución lagrangiana será la solución óptima del problema original, esto es $Lz^* = z^*$.

2.3.2 PROPIEDAD DE INTEGRALIDAD

Otra característica de un Relajación Lagrangiana, es que si en un modelo relajado se tiene la propiedad de Integralidad entonces el modelo relajado contiene la envoltura convexa en su espacio de solución ($Co\{x \in X | Cx \leq d\} = Co\{x \in X | Cx \leq d\}$) por lo que la solución al dual lagrangiano será exactamente igual a la relajación lineal del problema sin relajar.

2.3.3 DESCOMPOSICIÓN EN SUBPROBLEMAS

Esta propiedad refiere a la propiedad que puede tener un modelo de separarse en subproblemas al ser relajado, la separación del problema en subproblemas permite resolver el problema de manera separada e incluso paralela.

Estos subproblemas no necesariamente pueden ser fácilmente resueltos, solo si tienen la condición de integralidad, pero podrán tener una situación más favorable que el problema original

2.4 DESCOMPOSICIÓN LAGRANGIANA

La descomposición lagrangiana refiere a una manera de construir una Relajación Lagrangiana; la descomposición lagrangiana tiene la capacidad de dividir un problema en dos subproblemas, mediante una reformulación al modelo original. Se crea una nueva variable artificial llamada "variable layering." o "variable splitting", siendo esta nueva variable, una variable espejo (esto es una variable con el mismo valor que otra) de otra original del modelo, esta técnica de crear estructuras de problemas aprovechables a través de una variable artificial se estudia por Glover y Klingman (1988) [12].

No debemos confundir esto con la descomposición en subproblemas que se da al simplemente relajar una restricción del modelo dada anteriormente, ya que en la descomposición lagrangiana se obliga a la separación del problema en por lo menos dos subproblemas, la descomposición en subproblemas dada anteriormente, no hace uso de una variable artificial. Una descomposición lagrangiana puede también aprovechar la propiedad de descomposición en subproblemas si es posible.

A continuación se presenta una interpretación geométrica de la descomposición lagrangiana. La descomposición lagrangiana puede ser mejor que relajar cada restricción individualmente, además a diferencia de la Relajación Lagrangiana normal esta Relajación usa las envolventes convexas de cada conjunto de restricciones y no de solo uno solo (ver figura 2.2).

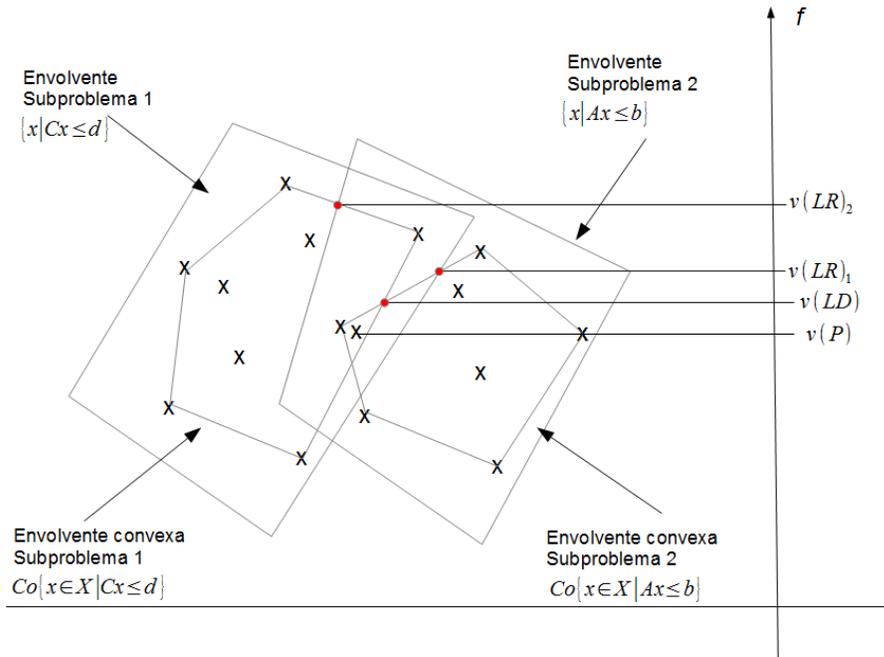


Figura 2.2: Interpretación geométrica de la descomposición lagrangiana

Tenemos un problema P definido con el siguiente modelo matemático

$$(P) \quad \min_x \{fx | Ax \leq b, Cx \leq d, x \in X\} \quad (2.9)$$

El problema anterior tiene dos grupos de restricciones; este modelo mediante la creación de las variables artificiales y , que asignamos en un grupo de restricciones nuevo con las variables x originales del modelo, y de ese modo se logra construir un nuevo modelo equivalente.

$$(P) \quad \min_{x,y} \{fx | Ax \leq b, Cy \leq d, x = y, x \in X, y \in X\} \quad (2.10)$$

Este nuevo modelo posibilita la división del problema en dos subproblemas al relajarse la restricción de igualdad $x = y$

$$(P') \quad \min_{x,y} \{fx + \lambda(y - x) | Ax \leq b, Cy \leq d, x \in X, y \in X\} \quad (2.11)$$

Agrupando los términos el problema queda como :

$$(P') \quad \min_x \{(f - \lambda)x | Ax \leq b, x \in X\} + \min_y \{\lambda y | Cy \leq d, y \in X\} \quad (2.12)$$

lo que nos deja ver dos modelos que se pueden resolver individualmente $P' = P1' + P2'$

$$(P1') \quad \min_x \{(f - \lambda)x | Ax \leq b, x \in X\} \quad (2.13)$$

$$(P2') \quad \min_y \{\lambda y | Cy \leq d, y \in X\} \quad (2.14)$$

La cota calculada de este modelo modificado puede ser mejor que relajar individualmente cada grupo de restricciones del modelo original, (ver figura 2.2)

2.5 RELAJACIÓN LAGRANGIANA MODIFICADA

Como extensión a la teoría clásica de la Relajación Lagrangiana tenemos la Relajación Lagrangiana modificada. Esta extensión trata de aprovechar las características que podría tener un modelo al ser dividido en dos problemas diferentes, se podría tener una solución fácil o de poder ser divididos en pequeños problemas. La Relajación Lagrangiana modificada utiliza variables artificiales; sin embargo a diferencia de la descomposición lagrangiana que introduce solamente un grupo de restricciones al modelo, la Relajación Lagrangiana modificada introduce varios grupos que intentan tener el mismo efecto.

La descomposición lagrangiana puede involucrar más cálculos debido a que puede contener un numero de multiplicadores tan alto como el de las variables involucradas ya que las variables artificiales son copias de la original. La Relajación Lagrangiana modificada utiliza un conjunto menor de restricciones y por lo tanto un número menor de multiplicadores.

Una explicación formal es la siguiente, dado el modelo matemático primal que se expresa como

$$(P) \quad z^* = \max_x \{cx | Dx \leq b, Ax \leq d, x \in U\} \quad (2.15)$$

introducimos la variable artificial y , y volvemos a escribir el modelo de la siguiente forma

$$(P) \quad z_M = \max_x \{cx | Dx \leq Dy, cx = cy, x \in X, y \in W\} \quad (2.16)$$

este modelo no es completamente idéntico al original pero los valores para $(x, y) = (x^*, x^*)$ son factibles en este modelo, donde x^* son los valores en x para la solución óptima global. Hay que hacer notar que $z^* \leq z_M$, y si sabemos además que el conjunto $X = \{x \in U \mid Ax \leq b\}$, y que el conjunto W implica la restricción $Dy \leq d$, esto hace que la solución óptima del modelo modificado sea factible en el modelo original. Por lo que sabemos una solución factible en ambos modelos podría ser menor o igual al óptimo así que $z^* \leq z_M$, pero como sabemos que no puede ser menor entonces $z^* = z_M$. Al hacer la Relajación de este modelo, llevando los grupos de restricción de enlace de variables a relajar nos queda un modelo que se puede ver como dos subproblemas; esto queda como

$$(P) \quad z_M(u, \eta) = \max_{x \in X} \{(1-v)cx - uDx\} + \max_{y \in W} \{\eta * cy - uDy\} \quad (2.17)$$

la cota $z_M(u, v)$ contiene apenas $m+1$ multiplicadores (un vector u y un escalar η) mientras que la descomposición lagrangiana utiliza una mayor cantidad de multiplicadores λ otra forma de ver la cota modificada es como la minimización de los términos complementarios de una relajación, como esto

$$(P) \quad \max_{x \in X} \{cx + u(d - Dx)\} - \min_{y \in W} \{u(d - Dy)\} \quad (2.18)$$

así que podemos observar a la cota modificada como un ajuste a la relajación al restarle una minimización de los términos complementarios

2.6 MÉTODOS PARA RESOLVER LA RELAJACIÓN LAGRANGIANA

MÉTODO DEL SUBGRADIENTE

Los métodos de gradientes son utilizados para obtener el máximo valor en una función cóncava, también llamados métodos de ascensos más pronunciados, buscan una solución óptima a un problema partiendo desde un punto inicial x^0 pasando por una secuencia de $\{x^0, x^1, \dots, x^n\}$, hasta eventualmente converger a una solución.

Cada nuevo punto en x^i es calculado con una función de ascenso; Guta[?] desarrolla detalladamente el tema de los métodos del subgradiente en relajaciones lagrangianas.

$$x^{n+1} = x^n + t_n \nabla f(x^n) \quad (2.19)$$

donde t_n es la longitud del paso de búsqueda y $\nabla f(x^n)$ es el vector gradiente de la función f en x^n ; esto es, el nuevo x está encaminado en la dirección de búsqueda dada por el gradiente(ver figura 2.3).

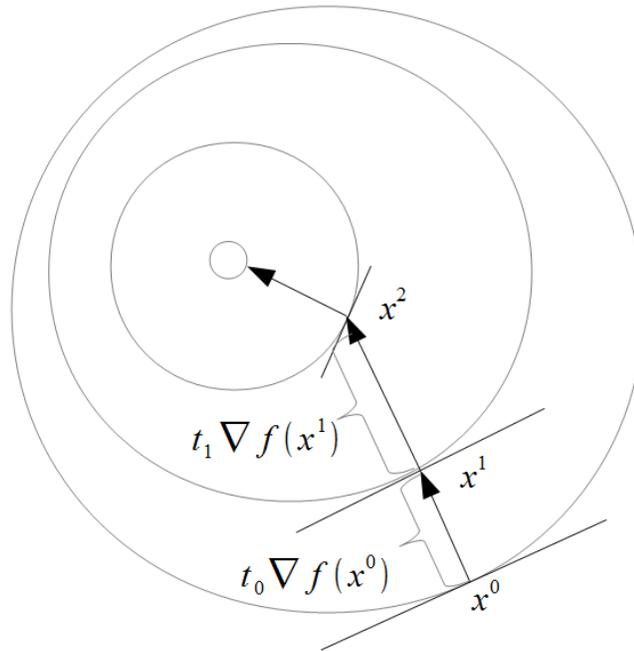


Figura 2.3: Ilustración del método del gradiente

En nuestro caso, la función no es diferenciable ya que es una función discreta y no es continua. El método del subgradiente es una adaptación del método de los gradientes, los gradientes son reemplazados por subgradietes. los métodos basados en los subgradietes no aseguran una solución exacta al problema, sin embargo son rápidos y fáciles de implementar.

Para aplicar el método del subgradiente debemos tener la función objetivo relajada.

$$L_k = \min_{x^k} \{f x^k + \lambda^k (A x^k - b)\} \quad (2.20)$$

donde λ es el vector de multiplicadores lagrangianos para el grupo de restricciones relajadas, x representa las variables solución y el índice k el número de iteración del subgradiente, para obtener e identificar el subgradiente se deriva la función objetivo con respecto a λ , así que nuestro gradiente sería:

$$\gamma^k = Ax^k - b \quad (2.21)$$

Se comienza calculando un valor de la función lagrangiana dado un vector λ arbitrario (aunque en la práctica se utilizan los valores duales de una Relajación lineal por ofrecer una mejor aproximación inicial, o simplemente un vector inicial donde todos los multiplicadores sean cero).

Una vez calculada esa solución de la función lagrangiana, se procede a calcular los subgradientes γ dada la fórmula 2.21, Podemos notar que la evaluación de subgradientes es parecido a la evaluación de las restricciones relajadas

Con el valor obtenido de la función lagrangiana y el cálculo de los subgradientes procedemos a calcular el tamaño del paso t_k con la siguiente fórmula.

$$t_k = \frac{\theta_k * (Ls - Lk(\lambda^k))}{\|\gamma_k\|^2} \quad (2.22)$$

donde Ls es el valor de una solución factible del modelo original, obtenido ya sea por un método heurístico o por el valor aproximado dado por un software comercial, L_k es la solución obtenida del modelo relajado en la iteración k y θ_k es un parámetro de ajuste del método del subgradiente (comúnmente inicia en 2). Si después de k iteraciones consecutivas con un valor fijo de θ el valor de la solución no mejora entonces se procede a modificar ese valor θ comúnmente $\frac{\theta}{2}$.

Para la siguiente iteración del subgradiente; el tamaño de paso multiplicado por el subgradiente γ correspondiente nos da la magnitud de cuanto deberán modificarse los multiplicadores para la siguiente iteración, la siguiente fórmula calcula el valor de los nuevos multiplicadores.

$$\lambda_{k+1} = \lambda_k + t_k * \gamma^k \quad (2.23)$$

Este método es el llamado subgradiente puro, pero existen otras variantes de este método que intentan hacer que el subgradiente converga más rápido, tales como

el método de deflexión del subgradiente, el método del subgradiente condicional utilizados para disminuir el zig-zag en la resolución del subgradiente puro.

MÉTODO DE BENDERS O DE GENERACIÓN DE RESTRICCIONES

Este método también llamado generador de restricciones, es también un método iterativo, en donde el modelo original es relajado para hacer más fácil su solución[?], se construye además un problema dual que tendría como variables los multiplicadores, las restricciones de este modelo dual se irían adicionando una a una en cada iteración con una función generadora de restricciones. El algoritmo inicia con una solución óptima del modelo relajado (podemos utilizar como multiplicadores iniciales λ_1 un valor aleatorio o simplemente cero).

$$(LR_{\lambda_1}) \quad \min_x \{fx + \lambda(Ax \leq b) | Cx \leq d, x \in X\} \quad (2.24)$$

Obtenemos un valor óptimo de solución inicial z^1 con x^1 y comenzamos la construcción del modelo dual.

Función objetivo del modelo dual

$$MP = \max_{\lambda} \theta \quad (2.25)$$

Agregamos una restricción con la función generadora de restricciones

$$\theta = fx^1 + \lambda(Ax^1 - b) \quad (2.26)$$

Obtendremos una solución al sistema, con un valor óptimo de solución que llamaremos θ^1 y multiplicadores igual a λ_2 . Con este nuevo valor de multiplicadores, regresamos al problema relajado original y resolvemos, dándonos una solución óptima con valor z^2 y x^2 , con lo que podemos agregar otra restricción al modelo dual con el nuevo valor repitiéndose el proceso entre modelo original y dual

La función generadora de restricciones para la iteración k sería expresada como

$$\theta = fx^k + \lambda(Ax^k - b) \quad (2.27)$$

el modelo dual queda expresado de la siguiente forma

$$MP^k = \max_{\lambda} \{\theta | \theta \leq f x^h + \lambda(Ax^h - b), h = 1, 2, \dots, k\} \quad (2.28)$$

Y el modelo relajado original podemos expresarlo en una iteración k como

$$LR_{\lambda_k} = \min_x \{f x + \lambda(Ax \leq b) | Cx \leq d, x \in X\} \quad (2.29)$$

El algoritmo de este método terminaría cuando los óptimos de la solución del modelo relajado y del dual sean iguales; esto es $LR_{\lambda_{k+1}} = MP^k$ quedando la solución del dual lagrangiano en ese valor final

El método de benders es un algoritmo exacto para encontrar el dual lagrangiano, pero es más costoso computacionalmente.

GENERACIÓN DE COLUMNAS

Análogo al método de generación de restricciones se tiene el método de generación de columnas (este método puede ser visto como una aplicación del método de Dantzing Wolfe [?][?][?], el cual consiste en reformular el problema de tal modo que las columnas del modelo correspondan a soluciones factibles de un subconjunto de restricciones del problema, en este se elige un pequeño conjunto de columnas y durante las iteraciones se van sumando columnas, el modelo para generación de columnas resulta podemos obtenerlo del modelo dual

$$(LR^*) \quad \max_{\lambda \geq 0} \{ \min_x \{ f x + \lambda(Ax - b) | Cx \leq d, x \in X \} \} \quad (2.30)$$

que expresado como una envolvente convexa queda

$$(LR^*) \quad \min_x \{ f x | Ax \leq b, x \in \text{conv} \{ x \in X | Cx \leq d \} \} \quad (2.31)$$

Para utilizar el método de generación de columnas debemos identificar las restricciones de acoplamiento $Ax \leq b$, las cuales se combinarán con una solución Ax^k y una nueva variable μ_k , tal que $\sum_{k \in K} \mu_k(Ax^k) \leq b$, partiendo del hecho de que

se trata de una combinación convexa $\sum_{k \in K} \mu_k = 1$, el modelo completo queda como:

$$(LR^*) \quad \min_x \left\{ \sum_{k \in K} \mu_k (f x^k) \mid \sum_{k \in K} \mu_k (A x^k) \leq b, \sum_{k \in K} \mu_k = 1, \mu \geq 0 \right\} \quad (2.32)$$

Una elección adecuada para x^k esta en elegir entre cualquiera de todos los puntos de la envolvente convexa o solo de sus puntos extremos $x \in Co\{x \in X \mid Cx \leq d\}$, para obtener los valores de la variable original se tiene $x = \sum_{k \in K} \mu_k x^k$ con $\sum_{k \in K} \mu_k = 1$ and $\mu_k \geq 0$.

La separación de un problema en un problema maestro y un subproblema es equivalente a la separación de restricciones guardadas y las restricciones dualizadas. Las columnas generadas son soluciones de subproblemas enteros que tienen las mismas restricciones de los subproblemas lagrangianos; al igual que benders, es un método exacto para el cálculo del dual lagrangiano.

MÉTODOS COMBINADOS

Una descripción detallada de estos métodos puede encontrarse Hiriart-Urruty y Lemarechal [26], [25], [23], kiwiel [?] y frangioni [16] introdujeron una extensión a los métodos de subgradietes, llamados métodos combinados en los cuales la información pasada es recolectada para proveer una mejor aproximación a la función lagrangiana.

MÉTODOS HÍBRIDOS DE DOS FASES

En este método presentado en 1994 por Guignard y Zhu [22] se hace uso de dos métodos: el del subgradiente y el de generación de restricciones. En una primera fase se hace uso del subgradiente para ajustar los multiplicadores y al mismo tiempo las restricciones correspondientes a las soluciones conocidas de los subproblemas lagrangianos son adheridas al problema maestro LP; esto es, hacer uso del método de generación de restricciones en la segunda fase. También se puede usar generación de columnas en la segunda fase [17] el cual es descrito por Guignard y Freville.

EL ALGORITMO DE VOLUMEN VA

Este algoritmo propuesto por Barahona y Anbil(2000) [3] es una extensión mas al método del subgradiente, pero también puede ser visto como una forma aproximada de la descomposición de Dantzing-Wolfe.

Dado un problema de programación entera (IP) con dos restricciones (r1 y r2)

$$\begin{aligned} (IP) \quad & \min \quad cx \\ (r1) \quad & st : \quad Ax \geq b \\ (r2) \quad & \quad \quad Dx \geq d \\ & \quad \quad \quad x \in \mathbb{Z}_+^n \end{aligned}$$

minimizando la formulación

$$\begin{aligned} (IP) \quad & \min \quad cx \\ - \quad & st : \quad Ax \geq b \\ & \quad \quad \quad x \in X = \{x \in \mathbb{Z}_+^n, Dx \geq d\} \end{aligned}$$

su Relajación Lagrangiana con respecto a r1 puede quedar expresado con zona convexa de X

$$\begin{aligned} z^* \quad & \min \quad cx \\ & st : \quad Ax \geq b \\ & \quad \quad \quad x \in conv(X) \end{aligned}$$

La Relajación Lagrangiana de este modelo puede reformularse, haciendo de x una combinación convexa de x^i , haciendo esto podemos hacer una reformulación similar al esquema de Dantzing-Wolfe para agregación de columnas

$$\begin{aligned} \phi^* \quad & \min \quad cx \\ & st : \quad Ax \geq b \\ & \quad \quad \quad x \in conv(X) = \{x : x = \sum_i \mu x^i, \sum_i \mu = 1, \mu \geq 0\} \end{aligned}$$

El cual puede ser rescrito como:

$$\begin{aligned} \phi^* \quad & \min \quad \sum_i (cx^i) \mu_i \\ & st : \quad \sum_i (Ax^i - b) \mu_i \geq 0 \\ & \quad \quad \quad \sum_i \mu_i = 1 \\ & \quad \quad \quad \mu_i \geq 0 \end{aligned}$$

Este problema puede ser reformulado con el dual:

$$\begin{aligned} \phi^* \quad & \max \quad z \\ \text{st} : \quad & z + u(Ax^i - b) \leq cx^i \\ & u \in \mathbb{R}_+^n, z \in \mathbb{R} \end{aligned}$$

o también :

$$\begin{aligned} \phi^* \quad & \max \quad z \\ \text{st} : \quad & z \leq ub + (c - uA)x^i \\ & u \in \mathbb{R}_+^n, z \in \mathbb{R} \end{aligned}$$

La formulación con parámetro u es equivalente al lagrangiano dual $z = \phi(u)$, por ultimo podemos expresar el problema solo minimizando la parte de holgura $(c - uA)x^i$, de tal forma

$$\phi(u) = ub + \min\{(c - uA)x^i : x^i \in X, \forall i\} \quad (2.33)$$

El cálculo de los pesos μ se realiza con la estimación de ciertos volúmenes asociados a las caras activas en una solución óptima dual.

CAPÍTULO 3

ESTADO DEL ARTE

3.1 INTRODUCCIÓN

En este apartado se hará un recorrido de los problemas actuales e históricos del problema ruteo de vehículos, y de otros problemas en la investigación de operaciones, describiendo su relación con nuestro modelo

3.2 PROBLEMA DE VRP CON VENTANAS DE TIEMPO SUAVES

En este problema existen un numero de vehículos localizados en un deposito, los cuales tienen que satisfacer la demanda de bienes de un conjunto de clientes ubicados en distintas localizaciones geográficas. Los vehículos tienen una capacidad limitada y existe un intervalo (o ventana de tiempo) en el que los clientes deben ser satisfechos, además se debe retornar al depósito. A este modelo se le llama VRPTW pero existen otros casos donde se permite llegar fuera de la ventana de tiempo pero causando una penalidad, Valverde et al (2007)[6] realizan una investigación del uso de programación por metas para modelar el problema, donde el modelo intenta reflejar las penalidades por las violaciones a la ventana de tiempo, a través de variables de desviación. Este modelo permite evaluar las consecuencias por poner diferentes penalidades por las violaciones a las ventanas de tiempo, dependientes

de los clientes. También se han propuesto heurísticas como Eglese et al (2008) [18], los cuales propusieron una búsqueda tabú con memoria adaptativa para resolver el problema. Ellos también mencionan que este tipo de problemas puede contener al de ventanas de tiempo duras e incluso llegar a resolverlo con los adecuados coeficientes de penalización.

3.3 PROBLEMA DE VRP CON ENTREGAS DIVIDIDAS

Este tipo de ruteo también llamado SDVRP fue introducido en la literatura por Dror y Trudeau (1989) [14] quienes definieron algunas propiedades estructurales y propusieron una búsqueda local, Dror et al [13] formularon un modelo de programación entera, aplicaciones reales del problema han sido abordadas por Sierksma y Tijseen [36] que consideraron el problema de determinar la agenda de vuelos de helicópteros para el intercambio de personal en plataformas. Otro problema es presentado por Archetti et al [2] que consideraron un problema de recolección de residuos donde los vehículos tenía muy poca capacidad y los clientes una gran demanda; ellos también consideraron restricciones como ventanas de tiempo, priorización en los clientes y diferentes tipos de vehículos y residuos, ellos propusieron un heurística de búsqueda tabu ara resolver el problema.

3.4 PROBLEMA DE RUTEO EN MINAS

Entre los problemas de optimización de la industria minera, se encuentra el problema del transporte del mineral en minas subterráneas, el del manejo de la flota de vehículos de acarreo, en la práctica el supervisor asigna los operadores para los vehículos tomando en cuenta el estado de los recursos pero son los operadores de los vehículos los que toman las decisiones de ruteo de acuerdo a su experiencia Cordova et al 2004 [10]. Esta problemática ha sido tratada por Gamache et al(2005)[19], que intentan resolver el problema de ruteo implementando el algoritmo del camino

más corto pero con estrategia de vehículo por vehículo y de manera más integral en el 2006 Beaulieu y Gamache [4] estudian el problema de gestión de vehículos y desarrollan un algoritmo de enumeración basado en programación dinámica para el enrutamiento y programación de vehículos que se mueven solo en dos sentidos (bidireccionales) en una red de transporte con un solo carril; en esta nueva propuesta o extensión se intenta aprovechar el modo bidireccional de desplazamiento de los vehículos en los segmentos del ruteo resultando una más eficiente formulación. El vehículo comienza su recorrido desde un punto inicial hasta su destino y se trata de encontrar la mejor ruta y programación posible para cada vehículo de tal manera que haga su recorrido en el menor tiempo posible, tratando también de evitar conflictos entre las rutas de los vehículos como se ve en la figura 3.1, el algoritmo toma también en cuenta el movimiento bidireccional del vehículo (hacia adelante y hacia atrás) pero asegurando que el destino (punto de servicio) es alcanzado por un movimiento hacia adelante; en este tipo de ambientes (minas subterráneas) las rutas son muy limitadas y además solo permiten un vehículo a la vez, comúnmente se elige una única ruta, este problema ha sido clasificado por Peters et al [30] como uno de los más difíciles en los sistemas de vehículos guiados (AGV).

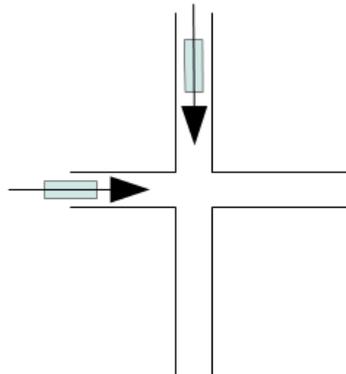


Figura 3.1: Ruteo en minas, evitar conflictos entre vehículos

3.5 PROBLEMA DE RUTEO DE CONTENEDORES

Este problema concierne a la entrega y recolección de bienes. En este problema de transportación el vehículo lleva un contenedor a un cliente donde este es vaciado o llenado; el vehículo parte de una terminal a un cliente con un contenedor donde este es vaciado o llenado y regresa a la terminal con el contenedor, Lo anterior establece que el contenedor puede estar vacío e ir con el cliente de recolección que lo llenará; o ir con el de entrega que vaciará para regresar con el contenedor vacío en lo que se llama triangulación de ordenes (de importación y exportación). Esta segunda alternativa es la más económica, Imai et al (2007)[24] utilizan un método heurístico basado en relajaciones lagrangianas para resolver el problema, siendo el método eficiente en instancias grandes, Reinhardt et al(2012)[32] diseña un modelo matemático para resolver un problema de este tipo, en el cual se reciben órdenes con ventanas de tiempo establecidas por los clientes y los contenedores de los vehículos deben ser del mismo tamaño para poder hacer una triangulación. No siempre será posible la triangulación, por lo que se requiere buscar la mejor combinación de viajes triangulados y simples. Podemos ver cómo serían los tipos de viajes posibles en la figura 3.2. 3.2.

Este modelo ejemplifica el caso de tener solo una terminal en donde salen los vehículos con sus contenedores, que pueden ser de distintos tamaños.

3.6 ABASTECIMIENTO DE COMIDA EN AVIONES

Un problema de este tipo se estudió por Sze (2012) et al[37]. Este problema trata del transporte de alimentos desde un centro (la cocina central) hacia las aeronaves. Se establecen restricciones para que los vehículos de suministro en cuanto a su capacidad para abastecer una limitada cantidad de aeronaves; otra restricción es el tiempo en que la comida tarde en salir y ser entregada a las aeronaves, estableciéndose un periodo de tiempo predeterminado desde la salida del centro de preparación la comida. Adicionalmente, los equipos de carga (personas) tienen un

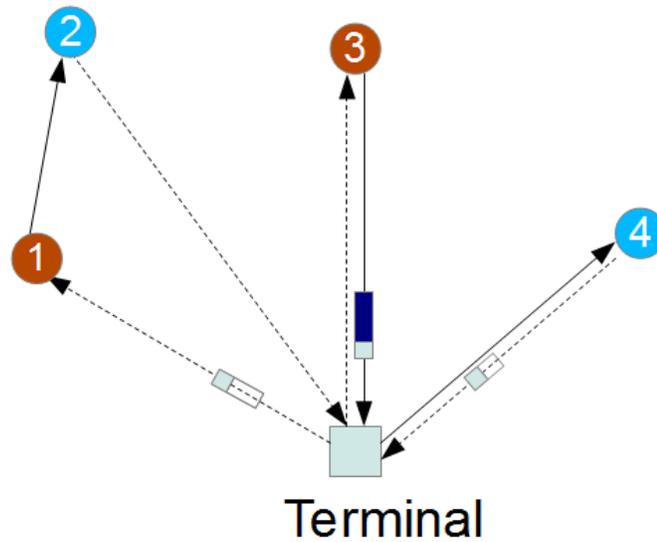


Figura 3.2: Problema de ruteo de contenedores

período de descanso. El objetivo fundamental del problema es satisfacer a todos los clientes (aeronaves) pero debido a las pequeñas ventanas de tiempo y a los distintos tipos de aeronaves, posiblemente más de un equipo se requiera para satisfacer la demanda del cliente (aeronave).

Otras restricciones son la capacidad del vehículo de carga. Todos los viajes deben ser agendados y tener su tiempo límite, siendo obligado que los clientes sean visitados una vez en la ventana de tiempo dispuesta para ellos. Tiempos de carga y descarga de paquetes de comida son incluidos en los tiempos de viaje por conveniencia.

Para este problema, por ser de gran tamaño y complejidad, se propuso una heurística de inserción de viajes rápida y que diera buenas soluciones

3.7 PROBLEMA DE RUTEO DE VEHÍCULOS DE EMERGENCIAS MEDICAS

El problema de ruteo de vehículos de emergencias es un problema de ruteo dinámico, donde los vehículos tienen ventanas de tiempo de servicio. El viaje de los vehículos es hacia un solo cliente para proporcionar el servicio necesario (consulta, traslado, etc.) y se debe volver al punto de origen. Existen prioridades de servicio y además es problema de naturaleza dinámica, va cambiando con el tiempo, conforme se van recibiendo llamadas de los clientes y conforme se les va dando el servicio, Rengifo et al(2012)[33] proponen un modelo matemático para este tipo de problema de ruteo para una empresa que provee este servicio de vehículos de emergencia en Medellín Colombia. En este modelo se manejan distintos tiempos de servicio, dependiendo de la naturaleza de la llamada hecha por el cliente. Se tiene el área total de servicio dividida en zonas con un número de posibles clientes y puntos donde pudieran ubicarse los vehículos. En este problema real existen dificultades para poder dar el servicio en los tiempos reglamentados (para mantener la calidad del servicio) por varias razones: parque vehicular limitado , dificultad para evaluar el impacto de una decisión de asignación de vehículos en las futuras asignaciones. En este problema real se intenta optimizar la operación de la flotilla de vehículos con que se dispone antes de acudir a una alta inversión por la compra de vehículos nuevos.

3.8 MAQUINAS PARALELAS

En este problema se tiene un conjunto de maquinas y un conjunto de trabajos a resolver , Zhilong y Powell (1996) [8] propusieron un método exacto de solución , basado en la descomposición de Dantzing Wolfe para un caso de máquinas paralelas que incluía la secuenciación de los trabajos y cuyo objetivo era minimizar el tiempo de la programación de trabajos.

Takashi et al(2007)[38] proponen una relajación lagrangiana para una variante de este tipo de problemas, en la cual se incluyen estaciones de trabajo donde se ubican las máquinas y los trabajos se procesan a través de estas estaciones pasando por un buffer intermedio entre estas para ser enteramente realizados. Se trabaja también con un modelo con ventanas de tiempo y capacidad limitada de buffer; el objetivo es minimizar el peso de las violaciones a las ventanas de tiempo. Ellos encontraron que trabajaba bien La relajación lagrangiana a las restricciones de capacidad y si el número de estaciones aumentaba podía usarse conjuntamente la restricciones de precedencia (las que definen en qué momento comienza un nuevo trabajo).

Una forma aproximada de ver el problema de VRP Starcase es hacer una analogía con el de máquinas paralelas , donde los vehículos podrían hacer la función de maquinas trabajando de manera paralela para hacer un trabajo.

CAPÍTULO 4

FORMULACIÓN MATEMÁTICA

4.1 INTRODUCCIÓN

El problema VRP-Starcase, es un problema de ruteo de vehículos en el cual los vehículos solo pueden visitar un cliente y regresar al depósito para recarga. Los vehículos pueden hacer solo un cierto número de viajes para satisfacer la demanda del cliente y también se tienen penalizaciones con el cliente por llegar antes o después de lo esperado (ventana de tiempo suave).

El objetivo es minimizar los costos por viajes realizados, vehículos utilizados y costos por multas; la solución dará como resultados: la flotilla de vehículos más adecuada, las rutas de los vehículos, y la secuenciación de los viajes a realizar por los vehículos.

El modelo planteado para resolver este problema es un modelo de programación lineal entera mixta. Podemos observar en este modelo matemático que no existen restricciones de subtour como en modelos clásicos ya que al no existir rutas entre clientes, los subtour son inexistentes.

4.2 MODELO MATEMÁTICO

Los parámetros del modelo son costos por viajes, utilización de vehículos, tiempos de viaje, horarios de entrega, demandas de clientes, multas, entre otros, que se listan en esta sección

Indices de arreglos

i .— Número de cliente (0 para depósito)

j .— Número de viaje, relacionado directamente con el vehículo, existiendo una cantidad de viajes

k .— Número de vehículo, no confundir con tipo de vehículo

Parámetros de vehículos

J_k Conjunto de viajes de vehículo k .

JM_k Máximo número de viajes de vehículo k .

C_{ik} Costo de viaje redondo desde el depósito hacia el cliente i del vehículo k .

q_k Capacidad del vehículo k .

t_{ik} Tiempo de viaje (ida) desde el depósito hacia el cliente i por el vehículo k .

f_k Costo de adquisición de vehículo k .

Parámetros de clientes

d_i .— Demanda del cliente i

Parámetros de ventanas de tiempo

E_i Tiempo inicial para comienzo de entrega de mercancía a cliente i .

L_i Tiempo final para entrega de mercancía a cliente i .

e_{ik} Multa por entrega de mercancía de vehículo k antes de tiempo inicial de cliente i .

l_{ik} Multa por entrega de mercancía de vehículo k después de tiempo final de cliente i .

M Constante que representa un valor muy grande.

las variables a utilizar son:

- x_{ijk} Variable de decisión, indica si un cliente i es visitado por un vehículo k , en el viaje j de ese mismo vehículo ($x_{ijk} = 1$ si es visitado).
- y_k Variable de decisión, indica si un vehículo es utilizado ($y = 1$ si es visitado).
- s_{jk} Tiempo inicial de viaje j del vehículo k .
- w_{ik}^- Tiempo de entrega de mercancía de vehículo k antes de tiempo inicial de cliente i .
- w_{ik}^+ Tiempo por entrega de mercancía de vehículo k después de tiempo final de cliente i .

La naturaleza de estas variables es:

$$x, y \in \{0, 1\} \text{ y } s_{jk}, w_{ik}^-, w_{ik}^+ \geq 0$$

Como tenemos variables binarias y continuas, tenemos un modelo de programación lineal entera-mixto

Se presenta el modelo matemático para el problema:

$$\begin{aligned}
 (0) \quad & \min : \sum_k f_k y_k + \sum_{i,j \in J_k, k} (c_{ik} x_{ijk} + e_{ik} w_{ijk}^- + l_{ik} w_{ijk}^+) \\
 (1) \quad & x_{0jk} + \sum_{i \geq 1} x_{ijk} = 1 \quad \forall j \in J_k, k \\
 (2) \quad & \sum_{j \in J_k, k} q_k x_{ijk} \geq d_i \quad \forall i \neq 0 \\
 (3) \quad & JM_k - \sum_{j \in J_k} x_{0,j,k} \leq y_k JM_k \quad \forall k \\
 (4) \quad & s_{j+1,k} \geq s_{jk} + \sum_{i,k} 2t_{ik} x_{ijk} \quad \forall j \in J_k, k \\
 (5) \quad & w_{ijk}^- \geq E_i x_{ijk} - (s_{jk} + t_{ik} x_{ijk}) \quad \forall i \neq 0, j \in J_k, k \\
 (6) \quad & w_{ijk}^+ \geq s_{jk} + t_{ik} x_{ijk} - L_i - M(1 - x_{ijk}) \quad \forall i \neq 0, j \in J_k, k
 \end{aligned}$$

A continuación se da una explicación de como funcionan y para qué sirven cada una de las restricciones y la función objetivo.

4.2.1 FUNCIÓN OBJETIVO

La función objetivo consta de tres partes, la primera calcula los costos de adquisición de los vehículos realmente utilizados, la segunda parte calcula el costo de los viajes realizados y la última los costos debidos a las multas por salirse de la ventana de tiempo.

Los costos de adquisición son superiores en gran proporción a los de transporación; en el caso de ventanas de tiempo duras, el costo de penalización es muy alto. El objetivo es minimizar los costos. Esto se puede ver en la siguiente figura 4.1.

$$\begin{array}{c}
 \min: \underbrace{\sum_k f_k y_k}_{\text{1.- Costos de adquisición}} + \sum_{i, k, j \in J[k]} \underbrace{\left(c_{ik} x_{ijk} + e_{ik} w_{ijk}^- + l_{ik} w_{ijk}^+ \right)}_{\substack{\text{2.-Costos} \\ \text{por viajes} \quad \text{3.-Costos por} \\ \text{penalizaciones}}}
 \end{array}$$

Figura 4.1: Descripción de función objetivo

4.2.2 RESTRICCIONES

El primer grupo de restricciones (llamaremos "asignación de viajes") obliga a que en un viaje se visite exactamente un cliente o se quede en el depósito; básicamente es sumar todos los clientes visitados por el vehículo k en el viaje j . La explicación se puede ver en la figura 4.2

La figura izquierda (visitando a un cliente), muestra que si se visita en el viaje j de un vehículo k a un cliente, se hace imposible visitar a otro para mantener la igualdad. La ilustración derecha (quedándose en el depósito), indica que se permaneció en el depósito (o cliente 0), no se puede visitar a otro cliente para mantener la igualdad de la restricción.

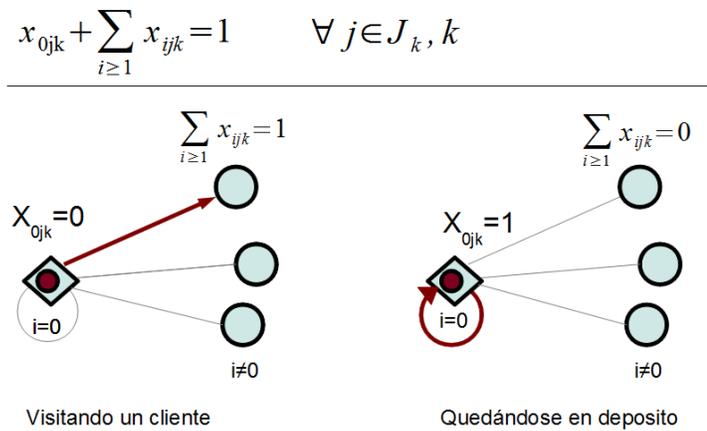


Figura 4.2: Descripción de restricción 1

El segundo grupo de restricciones (llamaremos "de capacidad") asegura la satisfacción de la demanda de los clientes; suma todas las entregas hechas al cliente realizadas por todos los vehículos en sus viajes. Una explicación puede verse en la figura 4.3.

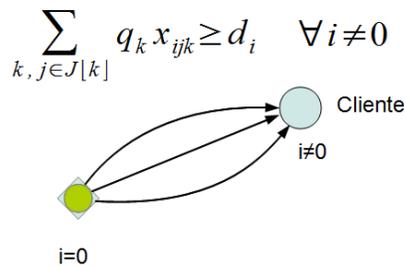


Figura 4.3: Descripción de restricción 2

La ilustración muestra los viajes realizados a un cliente (que no sea el depósito) donde las entregas deberán sumar al menos la demanda del cliente

El tercer grupo de restricciones está relacionado con la utilización de vehículos, establece cuales fueron los que realmente se utilizan, marcando como $y=1$ los vehículos que realmente son usados; está relacionada con la primer parte de la función objetivo que calcula los costos por adquisición de vehículos. Puede verse en la figura 4.4

La ilustración muestra los dos casos posibles, el caso en que se visita al menos

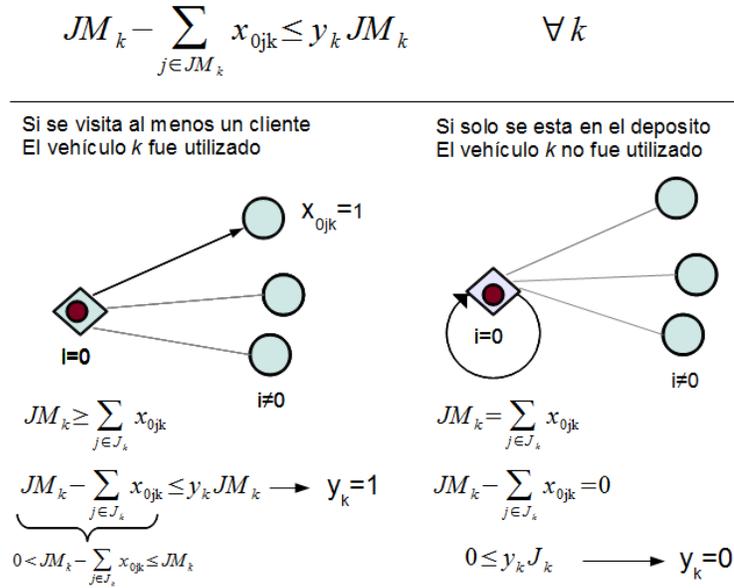


Figura 4.4: Descripción de restricción 3

un cliente, que obliga a marcar el vehículo como utilizado $y_k = 1$ y el caso en que todos los viajes fueron al depósito. Se marca el vehículo como no utilizado $y_k = 0$.

El cuarto grupo de restricciones establece la secuenciación de los viajes, obligando que un viaje de un vehículo solo pueda comenzar después de haberse terminado el anterior viaje del mismo vehículo, un viaje es de ida y vuelta por lo que el tiempo deberá ser multiplicado por dos para el cálculo del viaje redondo (ver figura 4.5).

$$s_{j+1,k} \geq s_{jk} + \sum_{i,k} 2 * t_{ik} x_{ijk} \quad \forall j \in J_k, k$$

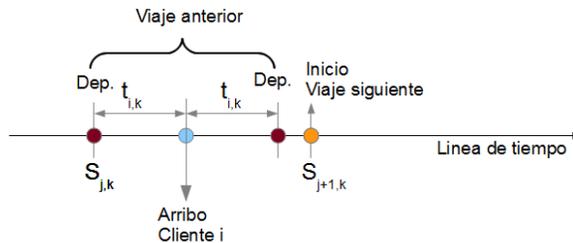


Figura 4.5: Descripción de restricción 4

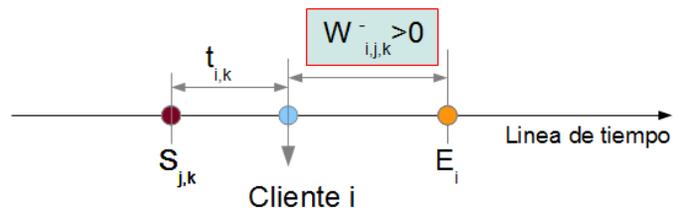
El quinto y sexto grupo de restricciones x son los que establecen los tiempos de violación, si es que los hubo. Estos dos grupos son muy similares al de secuenciación,

pero estos solo calculan al momento en que se llega al cliente.

El quinto grupo calcula el tiempo de violación si se llegó antes de lo permitido con el cliente, siendo 0 si se llegó después de la hora inicial permitida por el cliente (ver figura 4.6).

$$w_{ijk}^- \geq E_i x_{ijk} - (s_{jk} + t_{ik} x_{ijk}) \quad \forall i \neq 0, j \in J_k, k$$

Vehículo llega antes de la hora inicial de arribo a cliente i permitida



Vehículo llega después de la hora inicial de arribo a cliente i permitida (no hay multa)

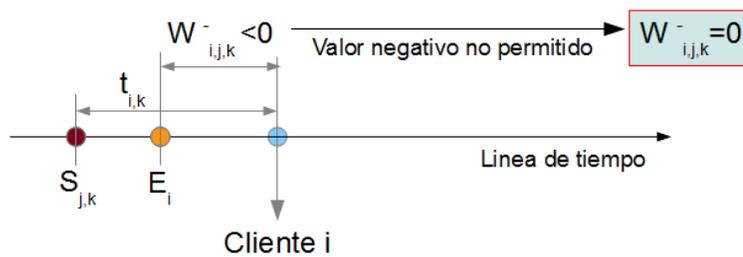


Figura 4.6: Descripción de restricción 5

El sexto grupo revisa si se llegó después de la hora final permitida, se hace uso de una constante M (con un valor muy grande) para calcular como cero los tiempos de violación en viajes que no se realizaron (ver figura 4.7).

$$w_{ijk}^+ \geq s_{jk} + t_{ik} x_{ijk} - L_i - M(1 - x_{ijk}) \quad \forall i \neq 0, j \in J_k, k$$

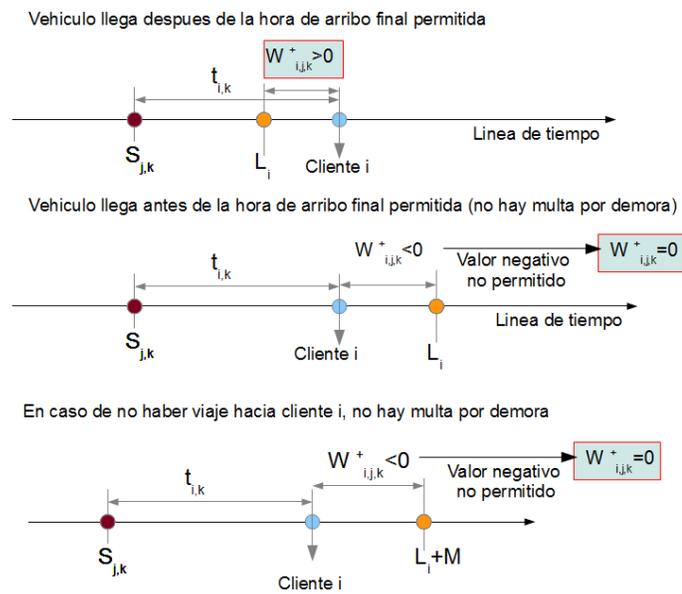


Figura 4.7: Descripción de restricción 6

CAPÍTULO 5

CONSTRUCCIÓN DE COTAS

5.1 INTRODUCCIÓN

En este capítulo desarrollamos los modelos relajados del modelo VRP-Starcase. se relajaran en este modelo todas las restricciones una por una y se pondrán en un formato estándar para facilitar las labores de programación; sobre estas relajaciones se aplicara el subgradiente el cual se explicará en detalle.

5.2 RELAJACIÓN 1ER GRUPO DE RESTRICCIONES

Separamos del modelo el grupo de restricciones que nos interesa, en este caso el primer grupo que es el de restricción de asignación de viajes:

$$(1) \quad x_{0jk} + \sum_{i \geq 1} x_{ijk} = 1 \quad \forall j \in J_k, k$$

Ponemos a este grupo de restricciones en un formato estándar para llevar a la función objetivo

$$1 - x_{0jk} - \sum_{i \geq 1} x_{ijk} = 0 \quad \forall j \in J_k, k \quad (5.1)$$

Este grupo de restricciones requiere una matriz de multiplicadores λ_{jk} correspondientes a cada pareja viaje-vehículo (jk) existente, para poder ser llevada como penalización a la función objetivo. La naturaleza de los multiplicadores es cualquier numero racional , sin importar el signo, debido a que es una restricción de igualdad

$$\lambda_{jk} \in \mathbb{R}$$

$$(LR_{\lambda_{jk}}) \quad \min_x \left\{ \sum_k f_k y_k + \sum_{i,j \in J_{k,k}} (c_{ik} x_{ijk} + e_{ik} w_{ijk}^- + l_{ik} w_{ijk}^+) \right. \\ \left. + \lambda_{jk} (1 - x_{0jk} - \sum_{i \geq 1} x_{ijk}) \right\} \quad (5.2)$$

La expresión anterior representa la función objetivo del modelo relajado, el modelo relajado puede entonces decirse que queda como función de los multiplicadores (función lagrangiana). Para obtener la cota debemos obtener el valor máximo de esta función.

$$Lz^* = \max_{\lambda} v(LR_{\lambda}) \quad (5.3)$$

5.3 RELAJACIÓN 2DO GRUPO DE RESTRICCIONES

El segundo grupo de restricciones a relajar es el de la satisfacción de la demanda de los clientes

$$(2) \quad \sum_{j \in J_{k,k}} q_k x_{ijk} \geq d_i \quad \forall i \neq 0$$

Este grupo de restricciones solo requiere un vector de multiplicadores λ_i correspondientes a cada uno de los clientes i en el sistema (excepto el índice 0 que es el del depósito)

$$(LR_{\lambda_{jk}}) \quad \min_x \left\{ \sum_k f_k y_k + \sum_{i,j \in J_{k,k}} (c_{ik} x_{ijk} + e_{ik} w_{ijk}^- + l_{ik} w_{ijk}^+) \right. \\ \left. + \lambda_i (1 - x_{0jk} - \sum_{i \geq 1} x_{ijk}) \right\} \quad (5.4)$$

Ponemos a este grupo de restricciones en un formato estándar (para facilitar la construcción de cotas), para llevarlo a la función objetivo, se cambia el sentido de la desigualdad para considerarla siempre negativa.

$$d_i - \sum_{j \in J_{k,k}} q_k x_{ijk} \leq 0 \quad \forall j, k \quad (5.5)$$

Continuamos con colocar esta expresión en la función objetivo con sus multiplicadores correspondiente, los multiplicadores solo podrán ser positivos $\lambda_i \geq 0$ debido

a la desigualdad \leq , con esto se mantiene que el valor dado de la función lagrangiana sea menor al óptimo global.

$$(LR_{\lambda_{jk}}) \quad \min_x \left\{ \sum_k f_k y_k + \sum_{i,j \in J_{k,k}} (c_{ik} x_{ijk} + e_{ik} w_{ijk}^- + l_{ik} w_{ijk}^+) + \lambda_i (d_i - \sum_{j \in J_{k,k}} q_k x_{ijk}) \right\}$$

Este valor como ya se ha mencionado es menor o igual a la solución óptima global. Para obtener la solución al dual lagrangiano se obtiene el máximo de la función lagrangiana, pero en este caso debe cuidarse el sentido de los multiplicadores.

$$Lz^* = \max_{\lambda \geq 0} v(LR_{\lambda}) \quad (5.6)$$

5.4 RELAJACIÓN 3ER GRUPO DE RESTRICCIONES

El tercer grupo de restricciones es el de asignación o utilización de vehículos

$$(3) \quad JM_k - \sum_{j \in J_k} x_{0,j,k} \leq y_k JM_k \quad \forall k$$

Ponemos a este grupo de restricciones en un formato estándar para llevar a la función objetivo, en este grupo no se cambio el sentido de la desigualdad al ponerla en el formato estándar.

$$JM_k - \sum_{j \in J_k} x_{0,j,k} - y_k JM_k \leq 0 \quad \forall k \quad (5.7)$$

Continuamos con colocar esta expresión en la función objetivo con sus multiplicadores correspondientes, solo se requerirá un vector de multiplicadores λ_k uno por cada vehículo, los multiplicadores solo podrán ser positivos $\lambda_k \geq 0$ debido a la desigualdad \leq .

$$(LR_{\lambda_{jk}}) \quad \min_x \left\{ \sum_k f_k y_k + \sum_{i,j \in J_{k,k}} (c_{ik} x_{ijk} + e_{ik} w_{ijk}^- + l_{ik} w_{ijk}^+) + \lambda_k (JM_k - \sum_{j \in J_k} x_{0,j,k} - y_k JM_k - \sum_{j \in J_{k,k}} q_k x_{ijk}) \right\} \quad (5.8)$$

5.5 RELAJACIÓN 4TO GRUPO DE RESTRICCIONES

El cuarto grupo de restricciones es el de secuenciación de viajes

$$(4) \quad s_{j+1,k} \geq s_{jk} + \sum_{i,k} 2t_{ik}x_{ijk} \quad \forall j \in J_k, k$$

Pasamos a poner este grupo en el formato estándar, este grupo necesitara también una matriz de multiplicadores λ_{jk} un multiplicador por cada pareja jk .

$$s_{jk} - s_{j+1,k} + \sum_{i,k} 2t_{ik}x_{ijk} \leq 0 \quad \forall j \in J_k, k \quad (5.9)$$

Los multiplicadores serán reales mayores que cero debido al sentido \leq de la desigualdad, La función objetivo ya con el grupo de restricciones incorporado es:

$$(LR_{\lambda_{jk}}) \quad \min_x \left\{ \sum_k f_k y_k + \sum_{i,j \in J_k, k} (c_{ik}x_{ijk} + e_{ik}w_{ijk}^- + l_{ik}w_{ijk}^+) \right. \\ \left. + \lambda_{jk}(s_{jk} - s_{j+1,k} + \sum_{i,k} 2t_{ik}x_{ijk}) \right\} \quad (5.10)$$

5.6 RELAJACIÓN 5TO Y 6TO GRUPO DE RESTRICCIONES

Estos grupos establecen los tiempos de violación de la instancia, son realmente dos conjuntos pero por cuestiones de concepto se relajan de manera conjunta.

$$(5) \quad w_{ijk}^- \geq E_i x_{ijk} - (s_{jk} + t_{ik}x_{ijk}) \quad \forall i \neq 0, j \in J_k, k$$

$$(6) \quad w_{ijk}^+ \geq s_{jk} + t_{ik}x_{ijk} - L_i - M(1 - x_{ijk}) \quad \forall i \neq 0, j \in J_k, k$$

Estos conjuntos van a requerir dos matrices de multiplicadores una por cada grupo u_{ijk} para (5) y v_{ijk} para (6), se pasan a poner en formato estándar.

$$E_i x_{ijk} - w_{ijk}^- - (s_{jk} + t_{ik}x_{ijk}) \leq 0 \quad \forall i \neq 0, j \in J_k, k$$

$$s_{jk} - w_{ijk}^+ + t_{ik}x_{ijk} - L_i - M(1 - x_{ijk}) \leq 0 \quad \forall i \neq 0, j \in J_k, k$$

La función objetivo con los dos conjuntos de restricciones integradas en la tiene como resultado:

$$(LR_{\lambda_{jk}}) \quad \min_x \left\{ \sum_k f_k y_k + \sum_{i,j \in J_k, k} (c_{ik}x_{ijk} + e_{ik}w_{ijk}^- + l_{ik}w_{ijk}^+) \right\}$$

$$\begin{aligned}
& +u_{ijk}(E_i x_{ijk} - w_{ijk}^- - (s_{jk} + t_{ik} x_{ijk})) \\
& +v_{ijk}(s_{jk} - w_{ijk}^+ + t_{ik} x_{ijk} - L_i - M(1 - x_{ijk}))\}
\end{aligned}$$

5.7 MÉTODO DEL SUBGRADIENTE - DETALLADO

El método del subgradiente desarrollado para calcular el dual lagrangiano es el método del subgradiente normal(Kipp[?]) pero con una estrategia en el avance similar a la planteada por Caprara et al [7]. Se presenta a continuación el proceso completo de solución de una instancia de un modelo relajado en el que se incluye la parte del subgradiente.

PARTE 1.- Antes de entrar a la construcción del modelo relajado. Se calcula una solución factible del modelo original y se calcula un valor inicial para los multiplicadores de acuerdo a los valores de las variables duales en una relajación lineal.

1.- Cálculo de una solución factible para el modelo original. En este caso se usó el mismo solver (cplex) para hallar esta solución pero modificando los parámetros para obtener una respuesta rápida del solver (el solver se detiene apenas alcanza un GAP dado, opcionalmente se puede utilizar una heurística para el cálculo de esta solución factible.

2.- Cálculo de una relajación lineal en el modelo original. Para obtener sus duales (valores relacionados con las restricciones) y usarlos como multiplicadores iniciales se realiza una Relajación Lineal.

2.1.-Relajación Lineal

2.1.1.-Se relaja modelo original.

2.1.2.-Variables enteras o binarias, pasan a ser continuas.

2.1.3.-Se resuelve modelo relajado.

2.1.4.-Se asigna a los multiplicadores λ los valores duales de solución de modelo relajado.

Los multiplicadores lagrangianos λ pudieran también ser estimados simplemente con un valor aleatorio o puestos todos en cero, aunque la práctica ha mostrado

que una buena estimación son los valores duales de solución.

PARTE 2.- Construcción del Modelo relajado. Aquí se agregan solo las restricciones no relajadas al modelo y se construye además la función objetivo.

1.- Construcción de restricciones del modelo relajado

$Ax \leq b$ Grupo de restricciones que queda fuera del modelo

$Dx \leq d$ Grupo de restricciones que se agrega al modelo

2.- Construcción de función objetivo (modelo original)

$$z = Cx$$

En este momento el modelo relajado aun no agrega la penalización y queda de esta forma

$$P : \min z = Cx$$

Sujeto a:

$$Dx \leq d$$

$$x \in X$$

PARTE 3.- Construcción y solución del modelo lagrangiano

1.- Agregación a la función objetivo de la penalización del modelo. Los multiplicadores lambda calculados hasta el momento, se agregan a la función objetivo para completar el modelo relajado.

$$z = z + \lambda(Ax - b)$$

$$\text{objetivo} = \text{minimizar} \{z\}$$

Agregar objetivo al modelo

El modelo LR_λ relajado esta completo y listo para ser resuelto:

$$LR_\lambda : \min z = c * x + \lambda(Ax - b)$$

Sujeto a:

$$Dx \leq d$$

$$x \in X$$

2.- Se calcula la solución del modelo LR_λ . Esta solución será nuestra cota actual Lk del problema.

PARTE 4.- Subgradiente, cálculo de los nuevos multiplicadores con el método del subgradiente.

1.-Verificación de la solución. Si hubo mejora en cota con respecto a la mejor hasta el momento llamada Lm , pasar al paso 2; en caso de no haber mejora saltar al paso 3.

Comparar $Lm < Lk$

2.- En caso de mejora ($Lm < Lk$). Se cambia la cota Lm por la nueva cota mejor Lk . Se almacena el valor de los multiplicadores (con el objetivo de guardar información de la iteración donde hubo mejora para regresar a ese punto).

$$L_m = L_k$$

$$\lambda_{mejor} = \lambda_k$$

$$\gamma_{mejor} = Ax - b$$

$$\gamma_k = \gamma_{mejor}$$

Si se utiliza el subgradiente modificado se incrementa el valor de θ , y se da un valor inicial para el número de faltas para probar solo unas cuantas veces esta nueva θ

$$\theta = \theta * 1.5$$

$$faltas = faltas_{max}/1.5$$

3.-En este paso se penaliza el subgradiente y se incrementan las faltas

$$faltas = faltas + 1$$

Si la falta fue por tiempo se incrementa en 1 las faltas por tiempo

$$faltastiempo = faltastiempo + 1$$

Si no se ha pasado máximo de faltas por tiempo

Si se ha pasado el máximo de faltas

$$\theta = \theta * 2$$

$$\lambda_{actual} = \lambda_{mejor}$$

$$\gamma_k = \gamma_{mejor}$$

No ha pasado el límite de faltas

$$\gamma_k = Ax - b$$

Se ha pasado del límite de faltas por tiempo

Ir a paso 7 (Fin del algoritmo)

4.-En este paso se calcula el nuevo multiplicador λ

4.1.- Cálculo de $\|\gamma_k\|^2$

$$\|\gamma\|^2 = \sum \gamma_k^2$$

4.2.- Cálculo del tamaño del paso t_k

$$t_k = \frac{\theta^*(Ls - Lk(\lambda^k))}{\|\gamma_k\|^2}$$

4.3.- Cálculo de los nuevos multiplicadores

$$\lambda^{k+1} = \lambda^k + t_k \gamma_k$$

Pero si multiplicadores pertenecen a una restricción del tipo \leq :

$$\lambda^{k+1} = \min(\lambda^k + t_k \gamma_k, 0)$$

5.- Revisar el criterio de parada (iteraciones, límite de tiempo, convergencia , etc.)

6.- Retornar a la PARTE 3 - paso 2, Si no se cumplió el criterio de parada, esto implica recalcular la función objetivo con los nuevos multiplicadores.

7.- Fin del algoritmo, retorna la mejor cota calculada hasta el momento con sus multiplicadores respectivos.

Esta es una descripción comentada del proceso de cálculo del dual lagrangiano a través del subgradiente; en el capítulo 6 se muestran los seudocódigos de este proceso.

5.8 SOBRE EL GAP

Para medir el error relativo de una solución factible obtenida ya sea por un método aproximado (heurísticos, metaheurísticos , etc.) con el valor real del óptimo global de un problema, se mide obteniendo la relación entre la solución obtenida y la real, esta relación en un problema de minimización se expresa como:

$$\%GAP = 100 * \frac{Z_f - Z_o}{Z_f} \quad (5.11)$$

Donde:

Z_f .- Es una solución factible del problema.

Z_o .- Es la solución óptima al problema.

$\%GAP$.- Es el error relativo en porcentaje de nuestra solución

Este GAP es el mejor que se puede estimar, ya que expresa de manera precisa el error de nuestra solución con la verdadera solución. La explicación de la relación

de este GAP con el error absoluto de nuestra solución factible se puede ver en la figura 5.1

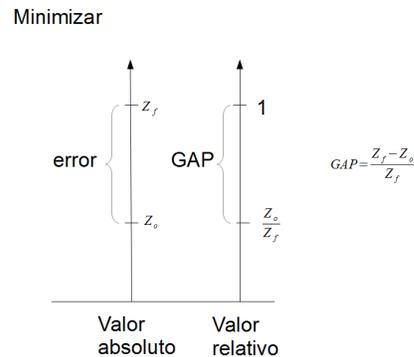


Figura 5.1: GAP entre solución factible y la óptima global

Los problemas a los que nos enfrentamos en programación lineal, no siempre podrán estar en relación con el valor óptimo del problema, ya que de hecho es lo que se intenta buscar. No se sabe de antemano cual es la solución óptima global, a menos que sea un problema de solución trivial. Un problema de programación entera, binaria o mixta necesitará otra forma de estimar el error; por ejemplo, en un problema de programación lineal entera, usando el método de branch and bound, se estima una solución factible al problema y esa solución entera estará relacionada con una solución de una Relajación Lineal, la relación entre esa solución entera y su solución lineal pudiera ser una estimación del error. Si el error no está en lo tolerado se haría un nuevo corte obteniendo una nueva solución factible y nueva solución lineal. Si quisiéramos obtener la solución óptima global tendríamos que reducir ese error hasta cero; la expresión siguiente calcula en por ciento ese error estimado.

$$\%GAP = 100 * \frac{Z_f - Z_l}{Z_f} \quad (5.12)$$

Donde:

Z_l .- Es la solución lineal del problema.

Este GAP expresa un error relativo de nuestra solución, no tiene la precisión del GAP estimado con una solución óptima global, es más parecido a un intervalo de

confianza, la explicación de la relación de este GAP con el error absoluto de nuestra solución factible se puede ver en la figura 5.2

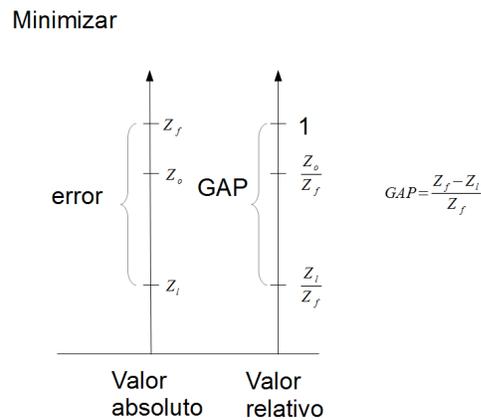


Figura 5.2: Descripción del GAP relativo a una solución lineal

La figura 5.2 muestra la relación que hay de nuestra solución factible con la solución óptima global, La solución óptima global está en un rango entre dos fronteras o cotas, una cota relajada del problema (la solución lineal) y la otra cota es la solución factible que obtuvimos. Entre más pequeño sea este rango más cerca esta nuestra solución de ser la solución óptima global, se puede decir que expresa que tan cerca esta nuestra solución de ser la verdadera, no que tan cerca esta de la solución real.

Para nuestra investigación usamos el mismo principio anterior: estimamos el error de nuestra solución relativo a una solución relajada, pero no utilizamos la relajación lineal, usamos la relajación lagrangiana, la expresión del GAP siguiente es una expresión general para cualquier tipo de relajación (lineal, subrogada, lagrangiana, etc.):

$$\%GAP = 100 * \frac{Z_f - Z_r}{Z_f} \quad (5.13)$$

Donde:

Z_r .- Es la solución relajada del problema

La figura 5.3 muestra el GAP deseado utilizando una relajación lagrangiana para estimar nuestra cota, se puede observar donde podría quedar la cota lineal con

respecto a nuestra cota lagrangiana y como la zona encerrada entre cotas pudiera ser menor.

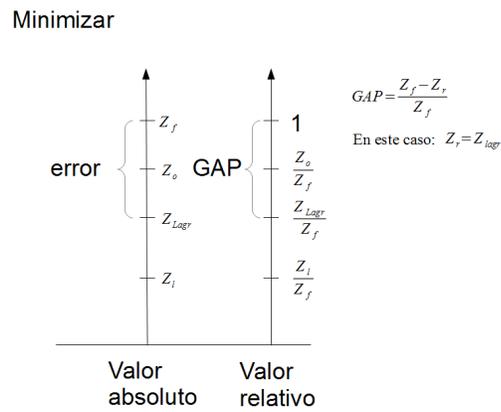


Figura 5.3: Descripción del GAP relativo a una solución lagrangiana

ión se seguiría Por ultimo, para el caso de maximización, las formulas serian:

Para el caso óptimo:

$$\%GAP = 100 * \frac{Z_o - Z_f}{Z_o} \tag{5.14}$$

Cuando se tiene una cota relajada:

$$\%GAP = 100 * \frac{Z_r - Z_f}{Z_r} \tag{5.15}$$

CAPÍTULO 6

IMPLEMENTACIÓN COMPUTACIONAL

6.1 INTRODUCCIÓN

En esta sección hablaremos de como se implementaron computacionalmente las relajaciones al modelo y las herramientas utilizadas para esto

6.2 GAMS

GAMS es un lenguaje de modelado algebraico, para construir modelos matemáticos de programación lineal y no lineal, es capaz de utilizar diferentes solvers para la resolución de problemas y modificar los parámetros de estos, su ventaja sobre un lenguaje de programación general es su facilidad de uso. No se requiere construir las matrices del modelo, solo construir las expresiones algebraicas que definen esas matrices facilitando mucho el modelado[34].

En esta investigación fue utilizado solo para verificar resultados de los modelos construidos con el lenguaje C++ y resueltos por el solver Cplex.

6.3 ILOG CPLEX OPTIMIZATION STUDIO

Ilog Cplex Optimization Studio es un conjunto de herramientas de software para resolver problemas de optimización. Una de las principales herramientas es un solver llamado Cplex. Este programa contiene una implementación del método sim-

plex hecha en lenguaje C, aunque hoy soporta otros métodos de optimización. El solver Cplex es un solver de alto rendimiento para problemas de Programación Lineal (LP), Programación Entera Mixta (MIP) y problemas de Programación Cuadrática (QP/QCP/MIQCP/MIQP) El solver Cplex ofrece varios algoritmos para resolver problemas de Programación Lineal, pudiendo elegir entre el algoritmo simplex primal, el algoritmo de puntos interiores entre otros. El solver Cplex además tiene un presolver que ayuda a reducir el tamaño de los problemas a ser resueltos Para la solución de problemas de Programación Entera Mixta, se provee de una implementación del algoritmo de branch and bound, el cual utiliza modernas técnicas como planos de corte y heurísticas para encontrar soluciones enteras. Ilog Cplex Optimization Studio ofrece adicionalmente herramientas para el modelado de programas, tales como un lenguaje propio de optimización llamado "Optimization Programming Language" (OPL) y librerías de componentes, que ofrecen interfaces para usar el solver en distintos lenguajes de programación.

6.4 LIBRERÍA CPLEX

La librería proporcionada para Cplex[1] provee lo necesario para usar todas las características del solver en distintos lenguajes de programación la cual nos permite usar el solver e interactuar con los parámetros de este, algo muy difícil con la consola de comandos . Se uso esta librería para resolver instancias y sus relajaciones, además de calcular duales, valor final de las restricciones ya que posee un evaluador de expresiones algebraicas (útil para el cálculo de subgradientes). La librería implementa características de modelado algebraico, muy parecido a GAMS, esto quiere decir que se pueden construir expresiones algebraicas para definir el modelo. La librería de Cplex para C++ contiene un conjunto de clases con las que definimos, variables, restricciones, modelos etc.. A continuación se listarán las más importantes:

- IloNumArray. Esta clase define arreglos de datos numéricos. En C++ es un arreglo de valores tipo float o double, su uso es opcional ya que pueden utilizarse

los tipos de datos proporcionados por C++

- `IloIntVar`, `IloNumVar`, `IloBoolVar`. Con estas clases definimos las variables enteras, continuas y binarias, asociadas a estas están las clases `IloIntVarArray`, `IloNumVarArray`, `IloBoolVarArray` que definen arreglos de variables, enteras, continuas y binarias respectivamente.
- `IloArray`. Es una plantilla que permite hacer arreglos de clases, es básicamente utilizada para hacer arreglos de arreglos variables (o matrices)
- `IloExpr`. En esta clase se guardan las expresiones algebraicas, con las que construiremos después las restricciones, también tiene una clase asociada llamada `IloExprArray` para definir arreglos.
- `IloRange`. Esta clase se utiliza para definir las restricciones del modelo, es útil para consultar los valores de las variables duales del problema. Tiene una clase asociada llamada `IloRangeArray` para definir arreglos de esta.
- `IloObjective`. Esta clase define la función objetivo del modelo.
- `IloModel`. Esta clase define todo nuestro modelo, es donde se agregan las restricciones y el objetivo.
- `IloCplex`. Esta clase es la que realiza el trabajo de solución del modelo, a través de esta se define el tipo de solver a utilizar, el tiempo máximo que se dejará correr la instancia del modelo y también guarda otras funciones como exportar el modelo a un archivo o importarlo entre otras.
- `IloEnv`. Esta clase es la que guarda toda la información del modelado y la que maneja la memoria que se vaya utilizando. Cuando se construye un modelo, todas las clases instanciadas deberán hacer referencia a ésta.

6.5 LENGUAJE C++

Aunque la librería de Cplex nos provee de muchas herramientas no es capaz por sí sola de generar instancias específicas, generar reportes de resultados amigables etc. para eso utilizamos C++. A continuación se listarán las funciones programadas en C++

- Creación de instancias simuladas. Se genera un archivo de datos con las matrices necesarias para procesar la instancia.
- Generación de reportes de instancias, de datos de entrada, de salida (soluciones modelo originales y modelos relajados)
- Construcción de modelos originales y relajados usando la librería de cplex.
- Implementación del subgradiente usando librerías de Cplex para el cálculo de expresiones del modelo.

El compilador utilizado fue el compilador de C++ para visual studio (Microsoft Visual C++ 9.0 Compiler), junto con el editor QtCreator

6.6 ARCHIVOS DE DATOS

Un archivo de datos es generado cuando se simulan instancias, el cual contiene el nombre del archivo, tipo de modelo, cuantas matrices tiene el archivo, y todas las matrices de la instancia, iniciando con su descripción (nombre de matriz y dimensiones). Este archivo es básicamente un archivo de matrices, es un archivo de texto que puede editarse en un bloc de notas.

6.7 ARCHIVOS DE SALIDA

No existe un solo archivo de salida, existen archivos de salida para la solución de instancias sin relajación y otros para instancias relajadas.

Para los modelos sin relajación tenemos:

- Archivo de reporte de solución. Éste es una interpretación de la solución dada por el software, contiene las agendas de los vehículos, de clientes, cuantos vehículos fueron utilizados, etc.
- Archivo de reporte de solución en formato matricial. Contiene los valores de las variables.

Para los modelos relajados

- Archivo de multiplicadores , generados durante el proceso del subgradiente
- Archivo de reporte de solución de la relajación, que contiene el valor final de la relajación (la cota lagrangiana), las características de cada iteración del subgradiente y los parámetros utilizados para el subgradiente.

6.8 ALGORITMOS

La parte más importante de la implementación computacional son cinco funciones: (1) la que implementa la resolución de modelos, considerando como criterio de parada el GAP,(2) una función que implementa la relajación lineal obteniendo los duales del grupo de restricciones relajado,(3) una función que construye los modelos, considerando que grupos de restricciones se relaján y cuales,(4) una función para construir la función objetivo y (5) la función propia del subgradiente, combinaciones de éstas forman las demás funciones de la implementación

Los pseudocódigos dados a continuación puedan dar una idea básica de la implementación, la implementación en C++ contiene más procesos intermedios no primordiales para el funcionamiento del programa pero útiles para el control de errores y crear los reportes

El siguiente pseudocódigo es una función para obtener una solución factible del modelo sin relajar

```
//OBTENER SOLUCION FACTIBLE
//GAP  - es el criterio de parada del solver (0 solucion global optima
)
//gr   - vector grupos de restricciones a relajar (0 sin relajacion)
//modelo - modelo a resolver
Solucion_factible(GAP)
begin
    gr<-Iniciar_sinrelajacion
    Cargar_datos (instancia)
    Dimensionar_variables
    Construir_restricciones(gr,&model)
    Construir_funcionobjetivo(&model)
    z<-resolver(GAP,model)
    retornar z;
end;
```

Este pseudocódigo, obtiene la relajación lineal del modelo y marca los grupos de restricciones donde queremos obtener los duales. Se considera que se dispone de los parámetros y las variables que son continuas, se retorna z y se modifica el valor de los multiplicadores U.

```
//OBTENER RELAJACION LINEAL
//GAP es el criterio de parada del solver
//gr son los grupos de restricciones a relajar (0 sin relajacion)
//U son los multiplicadores a inicializar
relajacion_lineal(gr,&U)
begin
    linealizar_variables
    Construir_restricciones(gr,&model)
```

```

    Zobj<-Construir_funciónobjetivo()
    //se agrega función objetivo a modelo para minimizar
    model->Objetivo(Zobj,minimiza)
    z<-solver(model)
    U<-obtenerduales(model,gr)
    retornar z;
end

```

El siguiente pseudocódigo es el de creación de grupo de restricciones. Considerando que si se relaja se adhiere a una lista de expresiones la restricción relajada. En esta implementación las matrices de multiplicadores se interpretan simplemente como una lista o vector, la lista de multiplicadores coincide con la lista de expresiones al momento de su utilización.

```

//Construir restricciones
//gr son los grupos de restricciones a relajar (0 sin relajacion)
//model modelo de la instancia
//lexpr lista de expresiones relajadas
Construir_restricciones(gr, &model)
begin

    Para i en Grupo 1
        expr <- Construir_expresion(i)
        Si gr[0]=0
            //relajar
            lexpr->add(expr)
        Si no
            rst <- (expr = 0)
            model->add(rst)
        Fin
        ..... // un ciclo diferente para cada grupo de restricciones
end test;

```

Seudocódigo para construir la función objetivo primal. Básicamente es construir la expresión algebraica del modelo y almacenar la expresión algebraica.

```

//Construir función objetivo
//gr son los grupos de restricciones a relajar (0 sin relajacion)
//model modelo de la instancia
//lexpr lista de expresiones relajadas
Construir_funciónobjetivo()
begin
    //Construye expresion z = cx
    Zobj← Construir_expresion
    retornar Zobj
end

```

Seudocódigo para construir la función objetivo relajada, Notar que se necesita la expresión de la función objetivos sin relajar ya que solo construye la parte relajada de la expresión algebraica.

```

//Construir función objetivo
//gr son los grupos de restricciones a relajar (0 sin relajacion)
//model modelo de la instancia
//lexpr lista de expresiones relajadas
Construir_función_objetivo_relajada(U, lexpr , Zobj , model)
begin
    Para cada i en lexpr
        MultaExpr←-U[i]* lexpr [ i ]
    Fin de i
    Zobj=Zobj+MultaExpr;
    model→Objetivo(Z, minimiza)
end ;

```

El seudocódigo del subgradiente presentado a continuación es el enfoque clásico. Se tiene una solución, se compara con la anterior, se actualiza la cota si esta es mejor, se modifican los multiplicadores de acuerdo a los subgradientes actuales del problema, si la cota no mejoró se penaliza el subgradiente y si pasa del límite de faltas se modifica el factor θ del paso; esto es, se hacen los pasos más pequeños. Suponemos

que ya se obtuvo de alguna manera una solución del problema sin relajar, necesaria para calcular el tamaño del paso.

```

Subgradiente(Lk,&U,&aprox)
begin
  Si (Lk>Lm)
    Lm<-Lk
    Umejor=U
  No
    multa<-multa+1
    Si multa=faltamax
      theta=theta/2
      U=Umejor
    Fin Si

  ro<-calcular_subgradientes_ro;
  tk<-calcular_tamaño_paso(ro,theta,Lf,Lm);
  Uant=U;
  U<-U+ro*tk
  aprox<-calcular_aprox(U,Uant)
end

```

El seudocódigo que se presenta a continuación es una pequeña modificación al subgradiente clásico. Es una estrategia de progreso que no modifica la forma de calcular el paso, solo contempla fallas por tiempo, esto es cuando la iteración tarda demasiado, se penaliza y no se modifica la cota. Se calculan los nuevos multiplicadores y la penalización por no mejorar cota sigue existiendo también pero además si hay mejora en cota se premia el factor θ de los multiplicadores (aunque se acorta la exploración para esta θ). Esta modificación se utiliza en esta tesis.

```

Subgradiente(tiempo_iter, Lk,&U,&aprox)
begin
  Si (Lk>Lm y tiempo_iter<tiempo_maxiter)
    Lm<-Lk

```

```

    Umejor=U
    theta=theta*1.5
    faltas=faltasmax/1.5
No
    multa<-multa+1
    Si tiempo_iter<=tiempo_maxiter
        faltatiempo<-faltatiempo+1
    Si faltatiempo<=faltatiempomax
        Si multa=faltasmax
            theta=theta/2
            U=Umejor
        No
            detener_porfaltatiempo <- verdad
    Fin Si
    Si detener_porfaltatiempo = falso
        ro<-calcular_subgradientes_ro;
        tk<-calcular_tamaño_paso(ro,theta,Lf,Lm);
        Uant=U;
        U<-U+ro*tk
        aprox<-calcular_aprox(U,Uant)
end
```

6.9 SIMULACIÓN

Para la simulación se utilizó un algoritmo sencillo, al que se le proporciona lo siguiente:

- La cantidad de clientes.
- La demanda mínima y máxima de los clientes.
- La ventana de tiempo de los clientes (igual para todos).
- La cantidad y tipos de vehículos.

- La proporción entre tipos de vehículos
- La capacidad del primer tipo de vehículo.
- Los costos de adquisición solo para el primer tipo de vehículo, los demás tipos estarán proporción de este.
- La proporción de capacidad, costos y tiempos entre tipos de vehículos (con respecto al primer tipo).
- Los costos mínimo y máximo que pudiera haber entre viajes entre vehículos y clientes.
- Los tiempos mínimo y máximo que pudiera haber entre viajes entre vehículos y clientes.
- El costo de penalización por demora (igual para todos los tipos de vehículos).
- El costo de penalización por llegar antes (igual para todos los tipos de vehículos).

El algoritmo trabaja creando valores aleatorios en los parámetros donde definimos un mínimo y un máximo, tales como demandas de clientes, costos de viaje y tiempos de viaje, para la cantidad de viajes se estima un valor igual para todos los vehículos de acuerdo a la demanda y capacidad del sistema, esto es procurar que se genere un área de soluciones factibles de la instancia.

CAPÍTULO 7

EXPERIMENTACIÓN COMPUTACIONAL

7.1 INTRODUCCIÓN

En este capítulo, se explicara cómo se crearon las instancias, la experimentación realizada en cada una de estas , los resultados de cada una de las experimentaciones y el tratamiento estadístico realizado con los resultados para inferir conclusiones (básicamente que % de GAP podemos esperar de la relajación lagrangiana).

7.2 CREACIÓN DE INSTANCIAS

Se crearon instancias de experimentación con un perfil semejante a una instancia real dada como ejemplo; las características de estas instancias fueron:

- La cantidad de vehículos sobrepasa a la de clientes.
- La capacidad de los vehículos es muy inferior a la demanda de los clientes.
- Los tiempos de viaje hacia un cliente i , es igual para cualquier vehiculo k .
- La consideración de una ventana dura (esto se traduce en el modelo a través de una penalización muy alta, igual para ambos casos de demora o llegada anticipada).
- Las ventanas de tiempo de los clientes son iguales para todos los clientes;

La experimentación con cotas lagrangianas se hizo con el siguiente diseño de experimentos:

- Se hicieron 4 tipos de instancias y 5 repeticiones para cada tipo (en total 20 instancias). Cada tipo de instancia supera en 10 vehículos al número de vehículos del tipo anterior. Se consideraron solo diez para todos los tipos de instancias. Se consideraron solo diez clientes para todas las instancias
- Se considero una ventana de tiempo 6:00 a 11:00 para todos los clientes en todos los tipos de instancias
- Solo se contemplan dos tipos de vehículos en la misma proporción, esto es , para 20 vehículos serán 10 de un tipo y 10 del otro. La proporción entre capacidad , costos de adquisición y de viaje entre tipos de vehículos es del doble, esto es, si el primer tipo tiene capacidad de 100 unidades el siguiente será de 200 unidades, lo mismo para costos de adquisición y de viaje.

El perfil de estas instancias es difícil de resolver para el solver Cplex. Los resultados ofrecidos por el solver en una hora no estiman un GAP aceptable para pensar que la solución es una buena solución; sin embargo, la hipótesis que se maneja en esta investigación es que un cálculo con otra cota (La proporcionada por la Relajación Lagrangiana) podría indicarnos que la solución dada por Cplex es mejor de lo estimado.

TIPO	I	K	J	DIM
TIPO A	10	30	6	1800
TIPO B	10	40	9	3600
TIPO C	10	50	12	6000
TIPO D	10	60	15	9000

Tabla 7.1: Tipos de instancias

Adicionalmente se hizo un reducido diseño de experimentos (con solo cuatro instancias una por cada tipo) para mostrar la dificultad que tienen CPLEX de llegar a la

solución óptima global del problema. Se calcularon soluciones y sus GAPS a 8 horas y 1 hora, los resultados se dan a continuación

TIPO	INST	GAP 8-horas	GAP 1-hora
TIPO A	INSTSR0	22.99 %	24.51 %
TIPO B	INSTSR3	43.49 %	44.11 %
TIPO C	INSTSR1	47.66 %	48.60 %
TIPO D	INSTSR3	52.41 %	52.41 %

Tabla 7.2: GAPS de soluciones Cplex para 8 horas y 1 hora

7.3 EXPERIMENTACION

El primer paso fue correr el solver durante una hora en todas las instancias y se recolectaron varios datos: la solución factible del problema hasta ese momento, el GAP de esa solución, con lo que se calculó su cota dual.

El segundo paso, fue para estas mismas instancias, calcular sus cotas lagrangianas, recolectándose adicionalmente al valor de la cota final, el tiempo, el número de iteraciones, los parámetros utilizados y el valor de la relajación lineal (RLin en tablas)

7.3.1 PARÁMETROS DEL SUBGRADIENTE

Para ajustar el valor del parámetro θ subgradiente se hicieron algunas experimentaciones previas, en estas experimentaciones se observó que los parámetros tradicionalmente usados del subgradiente no ofrecían una mejora en la cota, experimentaciones aleatorias previas mostraron la utilidad de utilizar un θ muy pequeño (de 0.001 e incluso 0.0001) para mejorar la cota, por lo que se definieron y usaron dos modos de subgradiente, el tradicional y una versión modificada para tratar de obtener buenas cotas.

el subgradiente modificado usado para la experimentación en esta investigación ofrece una estrategia que puede incrementar el valor de θ en caso de mejora con el propósito de aumentar con eso la capacidad de exploración del subgradiente a pesar de tener un θ muy pequeño

7.3.2 CALCULO DEL GAP

Para calcular el GAP relativo a la solución de Cplex y a la cota lagrangiana utilizamos la siguiente formula:

$$GAP_{rel} = \frac{Z_{cplex} - Z_{lagr}}{Z_{cplex}} \quad (7.1)$$

Donde:

GAP_{rel} .- Es el GAP calculado con la cota lagrangiana en lugar de la calculada por el solver

Z_{cplex} .- Es el valor factible dado por el solver, pudiendo ser o no el valor optimo global

Z_{lagr} .- Es el valor de la mejor cota lagrangiana obtenida por el método del subgradiente

7.4 RESUMEN RESULTADOS

Se obtuvieron resultados para los dos primeros grupos de restricciones (G0 y G1), en G0 solo se utilizo el modo básico del subgradiente ya que no se considero necesario usar el modificado ya que el modificado necesita que la cota muestre algún progreso en algún momento para poder incrementar θ , la experimentación en G0 no fue capaz de mejorar la cota en ningún momento, con la que inicio fue con la que termino, los demás grupos no pudieron simplificar el problema lo suficiente para llegar a una solución relajada en todas las instancias probadas, por lo que se puede decir que estos dos grupos restricciones son los "difíciles", es interesante observar que los últimos dos grupos (los de cálculo de penalización) a pesar de ser los que

proveen la mayor cantidad de restricciones no facilitan la solución del problema

Se muestran a continuación un resumen de las tablas del capítulo 9(Tablas 9.1,9.3,9.4,9.5). Se muestra el GAP calculado para las relajaciones realizadas para todas las instancias, agrupadas por tipo, grupo de restricción relajado y modo de subgradiente utilizado:

TIPO A 10x30				
	CPLEX	LAGRANGE GAP		
INST	GAP	G0	G1 BAS	G1 AV
INSTSR1.MTL	21.48 %	21.48 %	4.93 %	4.81 %
INSTSR2.MTL	18.07 %	18.07 %	4.11 %	4.15 %
INSTSR3.MTL	14.37 %	14.37 %	4.60 %	4.22 %
INSTSR4.MTL	11.83 %	11.83 %	2.47 %	2.61 %
INSTSR5.MTL	1.92 %	1.92 %	4.39 %	4.39 %

Tabla 7.3: GAPs Tipo A

TIPO B 10x40				
	CPLEX	LAGRANGE GAP		
INST	GAP	G0	G1 BAS	G1 AV
INSTSR1.MTL	38.82 %	38.82 %	13.52 %	13.04 %
INSTSR2.MTL	37.89 %	37.89 %	15.46 %	22.63 %
INSTSR3.MTL	38.48 %	38.48 %	17.35 %	20.69 %
INSTSR4.MTL	86.69 %	86.69 %	79.54 %	79.10 %
INSTSR5.MTL	41.92 %	42.00 %	18.67 %	18.13 %

Tabla 7.4: GAPs Tipo B

TIPO C 10x50				
	CPLEX	LAGRANGE GAP		
INST	GAP	G0	G1 BAS	G1 AV
INSTSR1.MTL	45.02 %	45.02 %	23.11 %	26.03 %
INSTSR2.MTL	49.28 %	49.28 %	35.36 %	29.02 %
INSTSR3.MTL	51.77 %	51.77 %	15.54 %	24.47 %
INSTSR4.MTL	70.88 %	70.88 %	49.78 %	43.70 %
INSTSR5.MTL	56.59 %	56.59 %	31.40 %	30.91 %

Tabla 7.5: GAPs Tipo C

El tipo D se dividió en dos tablas porque se hicieron varias relajaciones con diferentes parámetros de subgradiente

TIPO D 10x60				
	CPLEX	LAGRANGE GAP		
INST	GAP	G0	G1 BAS1	G1 AV1
INSTSR1.MTL	66.90 %	63.33 %	34.40 %	35.04 %
INSTSR2.MTL	63.57 %	54.25 %	63.95 %	63.95 %
INSTSR3.MTL	72.07 %	66.33 %	47.15 %	56.59 %
INSTSR4.MTL	82.81 %	83.26 %	72.95 %	75.46 %
INSTSR5.MTL	69.91 %	73.50 %	47.78 %	70.19 %

Tabla 7.6: GAPs Tipo D parte 1

TIP0 D 10x60				
	LAGRANGE GAP			
INST	G1 BAS2	G1 AV2	G1 BAS3	G1 AV3
INSTSR1.MTL	38.08 %	67.32 %	38.17 %	42.73 %
INSTSR2.MTL	50.47 %	50.05 %	50.47 %	53.83 %
INSTSR3.MTL	67.26 %	48.40 %	52.17 %	52.17 %
INSTSR4.MTL	70.92 %	72.06 %	72.94 %	72.06 %
INSTSR5.MTL	39.67 %	43.88 %	36.97 %	39.69 %

Tabla 7.7: GAPs Tipo D parte 2

7.5 SOBRE LA COTA LAGRANGEANA

En esta parte se calcula un estimado de la mejora en la cota con respecto a la ofrecida por Cplex. Seleccionando la mejor relajación de las probadas para cada instancia; esta estimación de mejora la pondremos en porcentaje con la siguiente formula:

$$Mejora_{rel} = 100 \left(\frac{GAP_{cplex} - GAP_{lagr}}{GAP_{cplex}} \right) \quad (7.2)$$

Donde:

$Mejora_{rel}$.-.- Estimado en porcentaje , de la mejora que se tiene de la cota con respecto a la de cplex

GAP_{cplex} .- Es la estimación de la desviación con respecto al óptimo global de la solución calculada por Cplex. la formula básicamente nos expresa que tanto mejora hubo de la cota con respecto a la que obtuvo Cplex; es una comparativa de porcentajes.

7.5.1 RESUMEN DE RESULTADOS

Se presenta a continuación un resumen de resultados, conteniendo para cada instancia el GAP dado por Cplex y el mejor GAP calculado con la cota lagrangiana. Los resultados se agrupan por tipos de instancias.

Instancias del TIPO A

INST	CPLEX GAP	Mejor GAP	Mejora
INSTSR1.MTL	21.48 %	4.81 %	77.61 %
INSTSR2.MTL	18.07 %	4.11 %	77.23 %
INSTSR3.MTL	14.37 %	4.22 %	70.63 %
INSTSR4.MTL	11.83 %	2.47 %	79.09 %
INSTSR5.MTL	1.92 %	1.92 %	0.00 %
		PROMEDIO=	60.91 %
		DESVIACION=	34.20 %

Tabla 7.8:

Instancias del TIPO B

INST	GAP	Mejor GAP	Mejora
INSTSR1.MTL	38.82 %	13.04 %	66.40 %
INSTSR2.MTL	37.89 %	15.46 %	59.18 %
INSTSR3.MTL	38.48 %	17.35 %	54.90 %
INSTSR4.MTL	86.69 %	79.10 %	8.76 %
INSTSR5.MTL	41.92 %	18.13 %	56.75 %
		PROMEDIO=	49.20 %
		DESVIACION=	23.02 %

Tabla 7.9:

Instancias del TIPO C

	CPLEX		
INST	GAP	Mejor	Mejora
INSTSR1.MTL	45.02 %	23.11 %	48.66 %
INSTSR2.MTL	49.28 %	29.02 %	41.13 %
INSTSR3.MTL	51.77 %	15.54 %	69.98 %
INSTSR4.MTL	70.88 %	43.70 %	38.35 %
INSTSR5.MTL	56.59 %	30.91 %	45.38 %
		PROMEDIO=	48.70 %
		DESVIACION=	12.53 %

Tabla 7.10:

Instancias del TIPO D

	CPLEX		
INST	GAP	Mejor	Mejora
INSTSR1.MTL	66.90 %	34.40 %	48.59 %
INSTSR2.MTL	63.57 %	50.05 %	21.27 %
INSTSR3.MTL	72.07 %	47.15 %	34.58 %
INSTSR4.MTL	82.81 %	70.92 %	14.36 %
INSTSR5.MTL	69.91 %	36.97 %	47.12 %
		PROMEDIO=	33.18 %
		DESVIACION=	15.25 %

Tabla 7.11:

El promedio de mejora de todas las experimentaciones y su desviación estándar son 48.00 % y 23.28 % respectivamente . En la mayor parte de los resultados se observaron mejoras significativas, tenemos en promedio un 48 % de mejora independientemente del tipo ,haciendo una estimación podemos estimar con un 70 % de confiabilidad en un 35.66 % la mejora que podríamos esperar de la relajación lagrangiana en este tipo

de perfil de instancias, para obtener este cálculo se usa la distribución normal de prueba con los promedios y desviaciones estándar de cada tipo. Para los modelos más grandes, podemos decir que la mejora esperada es menor, estando esa posible mejora por arriba del 35 % (Tipo D) sin embargo la reducción es más marcada ya que los GAPs estimados en estas instancias son más grandes, para obtener el valor de mejora esperado utilizamos la fórmula para transformar a una distribución normal estándar cuando la media no es cero.

$$Z = \frac{X - \mu}{\sigma} \quad (7.3)$$

Despejando X de la ecuación obtenemos la mejora esperada

$$X = Z\sigma + \mu \quad (7.4)$$

Donde Z es el valor obtenido del valor z para un área bajo la curva normal

En nuestro caso para estimar, sería ubicarse en 0.30 de área bajo la curva normal y dar el estimado con 70 % de probabilidad; esto podría representarse como $Z = Z_{0.30}$ donde $Z_{0.30} = 0.53$

En base a estos resultados (ver tabla 7.12) podemos decir que existe una mejora significativa mayor al 35 % sobre la cota proporcionada por Cplex, para el perfil de instancias investigado.

			MEJORA
	PROMEDIO	DESVIACION	ESPERADA
TIPO A	60.91 %	34.20 %	42.79 %
TIPO B	49.20 %	49.20 %	49.20 %
TIPO C	48.70 %	12.53 %	48.70 %
TIPO D	33.18 %	15.25 %	33.18 %

Tabla 7.12:

CAPÍTULO 8

CONCLUSIONES

En esta investigación se puede concluir que la utilización de las relajaciones lagrangianas puede resultar conveniente para estimar la calidad de una solución factible dada por Cplex para el problema de VRP Starcase. En instancias con un perfil complicado por su tamaño y estructura interna (instancias donde la cantidad de vehículos es mayor a la de clientes y la capacidad de los vehículos es pequeña comparándola con la demanda)

Se mostró a través de la experimentación que las soluciones ofrecidas por el solver están más cerca de la solución óptima global que la estimación dada por el solver Cplex

También se mostró una modificación a la estrategia de avance del método de subgradiente que puede trabajar mejor para este modelo , en la mitad de los casos hubo una mejor cota que la ofrecida por el método tradicional, ya que contempla la posibilidad de instancias donde si en una iteración del subgradiente demora demasiado tiene la capacidad de continuar con la siguiente iteración, el método simplemente rechaza esa instancia y ve la iteración como una penalización y continua con el cálculo para el siguiente subgradiente. Lo anterior indica que el problema tiene una sensibilidad muy grande a un cambio pequeño de parámetros, también ofrece una mayor posibilidad de exploración al no ir decreciendo siempre ya que premia cuando existe una mejora

Adicionalmente se implementó un modelo de programación que puede ser útil para relajaciones lagrangianas en otros tipos de modelos, ya que generaliza los mo-

delos como conjuntos de restricciones y con una función objetivo.

8.1 TRABAJO FUTURO

El trabajo futuro sera implementar un método heurístico para llegar a soluciones factibles a través de soluciones lagrangianas. También será implementar de manera más eficiente la descomposición lagrangiana junto con la cota modificada para este modelo y en general para otros tipos de problemas. Implementar y evaluar el método de benders para este tipo de problemas, además de iniciar la implementación del método de volúmenes.

CAPÍTULO 9

TABLAS

9.1 SOLUCIONES CPLEX

TIPO A—10x30					
Instancia	DIM	Objetivo Z	%GAP	COTA INF	Tiempo sec
INSTSR1.MTL	1800	543040	21.4840	426373.29	12651.6
INSTSR2.MTL	1800	525760	18.0691	430759.90	3600.55
INSTSR3.MTL	1800	545280	14.3657	466946.71	3600.2
INSTSR4.MTL	1800	549680	11.8251	484679.79	3600.63
INSTSR5.MTL	1800	433480	1.9224	425146.65	3600.16
TIPO B—10x40					
Instancia		Objetivo Z	%GAP	COTA INF	Tiempo sec
INSTSR1.MTL	3600	675520	38.8178	413298.00	3601.01
INSTSR2.MTL	3600	665720	37.8871	413498.00	3600.2
INSTSR3.MTL	3600	635320	38.4758	390875.55	3600.71
INSTSR4.MTL	3600	3280440	86.6923	436551.11	3600.26
INSTSR5.MTL	3600	701000	41.9242	407111.36	3600.25

Tabla 9.1: Tabla Soluciones Cplex

TIPO C—10x50					
Instancia		Objetivo Z	%GAP	COTA INF	Tiempo sec
INSTSR1.MTL	6000	477530	45.0233	262530.24	3600.96
INSTSR2.MTL	6000	588420	49.2845	298420.15	3600.3
INSTSR3.MTL	6000	599320	51.7654	289079.60	3600.3
INSTSR4.MTL	6000	842980	70.8795	245479.99	3600.34
INSTSR5.MTL	6000	544890	56.5900	236536.75	3600.9
TIPO D—10x60					
Instancia		Objetivo Z	%GAP	COTA INF	Tiempo sec
INSTSR1.MTL	9000	716000	66.8995	236999.58	3600.46
INSTSR2.MTL	9000	652280	63.5719	237613.21	3600.7
INSTSR3.MTL	9000	858480	72.0653	239813.81	3600.7
INSTSR4.MTL	9000	1466360	82.8128	252026.23	3600.43
INSTSR5.MTL	9000	892534	69.9134	268533.13	3600.45

Tabla 9.2: Tabla Soluciones Cplex - continuacion

9.2 PARAMETROS DEL SUBGRADIENTE

SUBGRADIENTE BASICO RELAJACION G1					
TIPO	THETA	FALTAS	ITER.	T. ITER.	T. MAX.
		MAX	MAX	(seg.)	(seg.)
TIPO A	0.001	18	400	120	1800
TIPO B	0.001	6	400	120	1800
TIPO C	0.001	6	400	120	1800
TIPO D BAS1	0.001	6	400	120	1800
TIPO D BAS2	0.0001	6	400	120	1800
TIPO D BAS3	0.0001	18	400	120	1800
SUBGRADIENTE MODIFICADO RELAJACION G1					
TIPO	THETA	FALTAS	ITER.	TIM. ITER.	TIM. MAX.
		MAX	MAX	(seg.)	(seg.)
TIPO A	0.001	18	400	120	1800
TIPO B	0.001	6	400	120	1800
TIPO C	0.001	6	400	120	1800
TIPO D AV1	0.001	6	400	120	1800
TIPO D AV2	0.0001	6	400	120	1800
TIPO D AV3	0.0001	18	400	120	1800

Tabla 9.3: Parametros subgradiente G1

SUBGRADIENTE BASICO RELAJACION G0					
TIPO	THETA	FALTAS	ITER.	TIM. ITER.	TIM. MAX.
		MAX	MAX	(seg.)	(seg.)
TIPO A	0.001	6	400	120	1800
TIPO B	0.001	6	400	120	1800
TIPO C	0.001	6	400	120	1800
TIPO D	0.001	6	400	120	1800

Tabla 9.4: Parametros subgradiente G0

9.3 SOLUCIONES - RELAJACION EN GRUPO G0

TIPO A 10x30					
Instancia	DIM	RLin	Lm	iter	tiempo
INSTSR1.MTL	1800	419104	426373	49	214.501
INSTSR2.MTL	1800	423371	430760	34	47.892
INSTSR3.MTL	1800	458357	466947	35	25.569
INSTSR4.MTL	1800	475104	484680	37	459.158
INSTSR5.MTL	1800	414432	425147	21	16.521
TIPO B 10x40					
Instancia	DIM	RLin	Lm	iter	tiempo
INSTSR1.MTL	3600	405545	413298	31	1813.71
INSTSR2.MTL	3600	406868	413498	52	1484.43
INSTSR3.MTL	3600	383449	390876	51	1539.82
INSTSR4.MTL	3600	431584	436551	24	1851.13
INSTSR5.MTL	3600	402807	406556	22	1815.38
TIPO C 10x50					
Instancia	DIM	RLin	Lm	iter	tiempo
INSTSR1.MTL	6000	262530	262530	25	1870.58
INSTSR2.MTL	6000	294237	298420	28	1919
INSTSR3.MTL	6000	285347	289080	25	1829.63
INSTSR4.MTL	6000	242367	245480	23	1829.76
INSTSR5.MTL	6000	233227	236537	37	1836.92

Tabla 9.5: Relajaciones G0

TIPO D 10x50					
Instancia	DIM	RLin	Lm	iter	tiempo
INSTSR1.MTL	6000	262530	262530	25	1870.58
INSTSR2.MTL	6000	294237	298420	28	1919
INSTSR3.MTL	6000	285347	289080	25	1829.63
INSTSR4.MTL	6000	242367	245480	23	1829.76
INSTSR5.MTL	6000	233227	236537	37	1836.92

Tabla 9.6: Relajaciones G0 continuacion

9.4 SOLUCIONES - RELAJACIÓN EN GRUPO G1

SUBGRADIENTE NORMAL

TIP0 A 10x30					
Instancia	DIM	RLin	Lm	iter	tiempo
INSTSR1.MTL	1800	419104	516271	169	1816.18
INSTSR2.MTL	1800	423371	504130	248	1700.92
INSTSR3.MTL	1800	458357	520217	199	1805.62
INSTSR4.MTL	1800	475104	536089	312	1808.38
INSTSR5.MTL	1800	414432	414432	23	8.393
TIP0 B 10x40					
Instancia	DIM	RLin	Lm	iter	tiempo
INSTSR1.MTL	3600	405545	584166	28	1909.51
INSTSR2.MTL	3600	406868	562768	24	1808.84
INSTSR3.MTL	3600	383449	525070	25	1895.45
INSTSR4.MTL	3600	431584	671061	19	1856.57
INSTSR5.MTL	3600	402807	570145	22	1837.39
TIP0 C 10x50					
Instancia	DIM	RLin	Lm	iter	tiempo
INSTSR1.MTL	6000	262530	367154	28	1916.23
INSTSR2.MTL	6000	294237	380330	18	1810.9
INSTSR3.MTL	6000	285347	506187	24	1834.57
INSTSR4.MTL	6000	242367	423375	18	1876.67
INSTSR5.MTL	6000	233227	373820	18	1826.06

Tabla 9.7: Relajaciones G1

TIP0 D BAS1 10x60					
Instancia	DIM	RLin	Lm	iter	tiempo
INSTSR1.MTL	9000	233957	469728	18	1822.34
INSTSR2.MTL	9000	235152	235152	16	1806.21
INSTSR3.MTL	9000	237840	453730	17	1904.14
INSTSR4.MTL	9000	249365	396711	18	1864.41
INSTSR5.MTL	9000	266048	466046	17	1858.73
TIP0 D BAS2 10x60					
Instancia	DIM	RLin	Lm	iter	tiempo
INSTSR1.MTL	9000	233957	443335	17	1861.24
INSTSR2.MTL	9000	235152	323090	16	1806.16
INSTSR3.MTL	9000	237840	281035	17	1892.86
INSTSR4.MTL	9000	249365	426346	18	1876.48
INSTSR5.MTL	9000	266048	538506	17	1839.49
TIP0 D BAS3 10x60					
Instancia	DIM	RLin	Lm	iter	tiempo
INSTSR1.MTL	9000	233957	442674	21	1840.84
INSTSR2.MTL	9000	235152	323090	17	1889.85
INSTSR3.MTL	9000	237840	410590	17	1887.49
INSTSR4.MTL	9000	249365	396862	18	1841.81
INSTSR5.MTL	9000	266048	562576	18	1899.73

Tabla 9.8: Relajaciones G1

9.5 SOLUCIONES - RELAJACIÓN EN GRUPO G1

SUBGRADIENTE MODIFICADO

TIP0 A 10x30					
Instancia	DIM	RLin	Lm	iter	tiempo
INSTSR1.MTL	1800	419104	516921	231	1806.99
INSTSR2.MTL	1800	423371	503966	226	1236.8
INSTSR3.MTL	1800	458357	522273	247	1806.11
INSTSR4.MTL	1800	475104	535359	342	1804.02
INSTSR5.MTL	1800	414432	414432	25	23.852
TIP0 B 10x40					
Instancia	DIM	RLin	Lm	iter	tiempo
INSTSR1.MTL	3600	405545	587406	26	1867.32
INSTSR2.MTL	3600	406868	515052	26	1814.52
INSTSR3.MTL	3600	383449	503864	27	1875.09
INSTSR4.MTL	3600	431584	685767	20	1886.3
INSTSR5.MTL	3600	402807	573901	24	1854.6
TIP0 C 10x50					
Instancia	DIM	RLin	Lm	iter	tiempo
INSTSR1.MTL	6000	262530	353237	26	1881.94
INSTSR2.MTL	6000	294237	417689	18	1894.3
INSTSR3.MTL	6000	285347	452642	22	1908.01
INSTSR4.MTL	6000	242367	474630	17	1813.24
INSTSR5.MTL	6000	233227	376469	17	1895.96

Tabla 9.9: Relajaciones G1

TIPO D AV1 10x60					
Instancia	DIM	RLin	Lm	iter	tiempo
INSTSR1.MTL	9000	233957	465121	17	1878.91
INSTSR2.MTL	9000	235152	235152	16	1807.24
INSTSR3.MTL	9000	237840	372698	17	1919.38
INSTSR4.MTL	9000	249365	359777	17	1906.07
INSTSR5.MTL	9000	266048	266048	16	1815.86
TIPO D AV2 10x60					
Instancia	DIM	RLin	Lm	iter	tiempo
INSTSR1.MTL	9000	233957	233957	16	1813.31
INSTSR2.MTL	9000	235152	325819	17	1820.87
INSTSR3.MTL	9000	237840	442986	17	1910.22
INSTSR4.MTL	9000	249365	409755	17	1846.96
INSTSR5.MTL	9000	266048	500930	17	1900.47
TIPO D AV3 10x60					
Instancia	DIM	RLin	Lm	iter	tiempo
INSTSR1.MTL	9000	233957	410083	18	1868.13
INSTSR2.MTL	9000	235152	301135	17	1874.66
INSTSR3.MTL	9000	237840	410599	17	1910.05
INSTSR4.MTL	9000	249365	409706	17	1815.11
INSTSR5.MTL	9000	266048	538283	17	1803.71

Tabla 9.10: Relajaciones G1

BIBLIOGRAFÍA

- [1] *ILOG CPLEX C++ API 9.0, Reference Manual*, ILOG S.A., 2003.
- [2] ARCHETTI, C., A. HERTZ y M. G. SPERANZA, «A tabu search algorithm for the split delivery vehicle routing problem», *Transportation Science*, **40**, págs. 64–73, 2005.
- [3] BARAHONA y ANBIL, «The volume algorithm: producing primal solutions with a subgradient method», *Mathematical Programming 87*, págs. 385–399, 2000.
- [4] BEAULIEU, M. y R. GAMACHE, «An enumeration algorithm for solving the fleet management problem in underground mines», *Computers and Operations Research*, v.6, n.33, págs. 1606–1624, 2006.
- [5] BENDERS, J., «Partitioning procedures for solving mixed-variables programming problems», *Numerische Mathematik*, **4**, págs. 238–252, 1962.
- [6] BERTSIMAS, D. y J. N. TSITSIKLIS, Athena Scientific, 1997.
- [7] BRUCE GOLDEN, E. W., S. RAGHAVAN, *THE VEHICLE ROUTING PROBLEM: LATEST ADVANCES AND NEW CHALLENGES*, Springer, 2008.
- [8] CALVETE, H. I., C. GALE, M.-J. OLIVEROS y B. SANCHEZ-VALVERDE, «A goal programming approach to vehicle routing problems with soft time windows.», *European Journal of Operational Research*, *177 (3)*, págs. 1720–1733., 2007.

-
- [9] CAPRARA, A., M. FISHETTI y P. TOTH, «A heuristic method for set covering problem», *Operations Research* 47(5), págs. 730–743, 1999.
- [10] CHEN, L., S. MATOBA y H. INABE, «Surrogate constraint method for optimal power flow», *IEEE Transactions on Power Systems* 13, págs. 1084–1089, 1998.
- [11] CHEN, Z.-L. y W. B. POWELL, «Solving Parallel Machine Scheduling Problems by Column Generation», *INFORMS J. on Computing*, **11**, págs. 78–94, 1999.
- [12] CLARKE, G. y J. WRIGHT, «Scheduling of vehicles from a central depot to a number of delivery points», *Operations Research*, págs. 568–581, 1964.
- [13] CORDOVA, F. M., L. QUEZADA, J. SEPÚLVEDA, V. BOLIVARES, L. ATERO y A. CONTRERAS, «A simulation model to enhance the operations of an underground mine», *International Workshop on Supply Chain Management Workshop on Supply Chain Management and Information System (SCMIS), Symposium Proceedings*, 2004.
- [14] DANTZIG, G. B. y J. H. RAMSER, «The Truck Dispatching Problem», *Management Science*, **6**, págs. 80–91, 1959.
- [15] DANTZIG, G. B. y M. N. THAPA, Springer, 1997.
- [16] DANTZIG, G. B. y P. WOLFE, *Decomposition Principle for Linear Programs*, 1960.
- [17] DINKEL, J. J. y G. A. KOCHENBERGER, «An implementation of surrogate constraint duality», *Operations Research* 26, págs. 358–364.
- [18] DOWER, F. y D. KLINGMAN, «LAYERING STRATEGIES FOR CREATING EXPLOITABLE STRUCTURE IN LINEAR AND INTEGER PROGRAMS», , págs. 165–181, 1988.
- [19] DROR, G. LAPORTE y P. TRUDEAU, «Vehicle routing with split deliveries», *Discrete Applied Mathematics* 50, págs. 229–254, 1994.

-
- [20] DROR, M. y P. TRUDEAU, «Split Delivery Routing», *Naval Research Logistics* 37, págs. 383–402, 1989.
- [21] ESPEJO y GALVAO, «O uso das relaxacoes Lagrangeana e surrogateem problemas de programação inteira», *Pesquisa Operacional*, v.22, n.3, págs. 387–402, 2002.
- [22] FISHER, M., «An Applications Oriented Guide to Lagrangian Relaxation», *INTERFACES*, 1985.
- [23] FLOOD, M. M., «The traveling-salesman problem», *Oper. Res.* 4, págs. 61–75, 1956.
- [24] FRANGIONI, A., «Dual-ascent methods and multicommodity flow problems», , 1997.
- [25] FREVILLE, A., M. GUIGNARD y S. ZHU, «Column generation and Lagrangean Relaxation: Two Powerful Related Tools for Integer Programming.», *OPIM Department Report 99?12?22*, 1999.
- [26] FU, Z., R. EGGLESE y LI, «A unified tabu search algorithm for vehicle routing problems with soft time windows», *Journal of the Operational Research Society*. 59, págs. 663–673., 2008.
- [27] GAMACHE, M., R. GRIMARD y P. COHEN, «A shortest-path algorithm for solving the fleet management problem in underground mines», *European Journal of Operational Research*, v.2 n.166, págs. 497–506, 2005.
- [28] GLOVER, F., «Surrogate constraint duality in Mathematical programming», *Operations Research Vol 23 No3*, 1975.
- [29] GUIGNARD, M., *TOP Volume 11*, Sociedad de Estadística e Investigación de Operaciones, 2003.
- [30] GUIGNARD, M. y ZHU, «A hybrid algorithm for solving Lagrangian duals in mixed-integer programming», *Proceedings of the VI Claió*, págs. 394–410, 1994.

- [31] GUTA, B., *Subgradient Optimization Methods in Integer Programming*, VDM, 2009.
- [32] HIRIART-URRUTY, J. y C. LEMARECHAL, «Convex analysis and minimization», *Algorithms II*, 1993.
- [33] IMAI, A., E. NISHIMURA y J. CURRENT, «A Lagrangian relaxation-based heuristic for the vehicle routing with full container load», *European Journal of Operational Research* 176, págs. 87–105, 2007.
- [34] KIWIEL, K. C., «Methods of descent for nondifferentiable Optimization», , 2000.
- [35] LEMARÉCHAL, C., «Nondifferentiable optimization: Subgradient and e-subgradient methods», *Lecture notes in optimization research*, 1975.
- [36] LEMARÉCHAL, C., «Bundle Methods in nonsmooth optimizations», *Vol. 3 of IIASA proceedings Series*, 1977.
- [37] LENSTRA, J. K. y A. H. G. R. KAN, «Complexity of vehicle routing and scheduling problems», *Networks*, 11(2), págs. 221–227, 1981, URL <http://dx.doi.org/10.1002/net.3230110211>.
- [38] LITVINCHEV, I., «Vehicle Routing in a Star Case Network», *First International Conference on Industrial, Mechatronics and Manufacturing Engineering*, 2007.
- [39] LITVINCHEV, I., S. RANGEL y J. SAUCEDO, «A Lagrangian bound for many-to-many assignment problems», , 2008.
- [40] MARTIN, R., *Large scale linear and integer optimisation: A uni?ed approach*, 1999.
- [41] PETERS, B., J. SMITH y S. VENKATESH, «A control classification of automated guided vehicle systems.», *International Journal of Industrial Engineering*, pág. 29?39, 1996.

- [42] PULLEN y WEBB, «A Computer Application to a Transport Scheduling Problem», *Journal of Computing* 10, págs. 10–13, 1967.
- [43] REINHARDT, L. B., S. SPOORENDONK y D. PISINGER, «Solving Vehicle Routing with Full Container Load an Time Windows», *Proceedings of Third International Conference, ICCL 2012*, 2012.
- [44] RENGIFO, J., J. ESCOBAR y M. GULNARA, «Diseño de un modelo matemático para el despacho de vehículos de emergencias médicas en Colombia», *Proceedings of CLAIO-SBPO 2012*, 2012.
- [45] ROSENTHAL, R. E., *GAMS Guide User*, GAMS Development Corporation, 2012.
- [46] ROSS, K., «Metaheuristics», *Lecture notes ISM2006*, 2006.
- [47] SIERKSMA y G. A. TIJSSEN, «Routing Helicopters for Crew Exchanges on Off-Shore Locations», *Annals of Operations Research, Vol. 76*, págs. 261–286, 1998.
- [48] SZE, S.-N., A. N. SUK-FONG y K.-L. CHIEW, «An insertion heuristic manpower scheduling for in-flight catering service application», , págs. 206–216, 2012.
- [49] TAKASHI, «Capacitated Vehicle Routing on Trees Operations Research», , 1991.
- [50] TOTH, P. y D. VIGO, *The Vehicle Routing Problem*, 1987.
- [51] WOLSEY, L. A., *Integer programming*, Wiley, 1998.
- [52] YEUN, L. C., W. R. ISMAIL, K. OMAR y ZIROUR, *M. Vehicle Routing Problem: Models and Solutions*, 2008.

FICHA AUTOBIOGRÁFICA

Lic. Luis Alfonso Infante Rivera

Candidato para el grado de Maestro en Ingeniería
con especialidad en Ingeniería de Sistemas

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Tesis:

COTAS LAGRANGIANAS PARA EL PROBLEMA DE RUTEO DE VEHÍCULOS EN RED TIPO ESTRELLA

Soy Luis Alfonso Infante Rivera, hijo de Lucia Rivera Alvarez y Jesus Infante Molina, naci en Monterrey el 19 de Noviembre de 1978, en el estado de Nuevo Leon, soy egresado en Licenciatura en Ciencias de la Computacion en el año 2010, de la facultad de Ciencias Fisico Matematicas que pertenece a la Universidad Autonoma de Nuevo Leon, posteriormente continue mis estudios en la Facultad de Ingenieria Mecanica y Electrica en la division de Posgrado en Ingenieria de Sistemas