



UNIVERSITAT DE VALÈNCIA

Departament d'Estadística e Investigació Operativa



UNIVERSITAT DE VALÈNCIA
REGISTRE GENERAL
ENTRADA

15 ABR. 2005

N.º 44987

HORA

OFICINA AUXILIAR N.º 17

El Problema General de Rutas con Viento

(Windy General Routing Problem, WGRP)

UNIVERSITAT DE VALÈNCIA
BIBLIOTECA CIÈNCIES

Nº Registre 23.565

DATA 12-07-05

SIGNATURA Biblioteca

TD-M 243

Nº LIBIS: i18974971 24cm.

b 16775405

Tesis doctoral
Isaac Plana Andani
2005

UMI Number: U603103

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI U603103

Published by ProQuest LLC 2014. Copyright in the Dissertation held by the Author.
Microform Edition © ProQuest LLC.

All rights reserved. This work is protected against
unauthorized copying under Title 17, United States Code.



ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

D. Angel Corberán Salvador, profesor titular del Departament d'Estadística i Investigació Operativa de la Universitat de València, y D. José María Sanchis Llopis, profesor titular del Departamento de Matemática Aplicada de la Universidad Politécnica de Valencia

CERTIFICAN:

que la presente memoria de investigación "**El Problema General de Rutas con Viento**" ha sido realizada bajo su dirección por Isaac Plana Andani y constituye su tesis para optar al grado de *Doctor en Ciencias Matemáticas*.

Y para que así conste, en cumplimiento de la normativa vigente, autorizan su presentación ante la Facultat de Matemàtiques de la Universitat de València para que puedan ser tramitadas su lectura y defensa públicas.

Burjassot, 14 de abril de 2005



Ángel Corberán Salvador



José María Sanchis Llopis

A Fani

Agradecimientos

En primer lugar, quiero dar las gracias al Ministerio de Ciencia y Tecnología y a la Generalitat Valenciana, que han financiado los proyectos “Un sistema de ayuda a la toma de decisiones en problemas de rutas de vehículos y localización de servicios” (TIC2000-1750-C06-01) y “Un sistema de ayuda a la toma de decisiones en problemas de rutas de vehículos y localización de servicios 2” (TIC2003-05982-C05-01), y “Soluciones avanzadas para problemas de rutas, distribución, transporte y otros” (GRUPOS03/189), respectivamente, bajo los cuales se ha realizado la labor investigadora recogida en esta memoria.

Muchas han sido las personas que, a lo largo de estos años, han contribuido de alguna manera a que esta tesis pudiera ser llevada a cabo. A todos ellos quiero dar las gracias, con mención especial a algunas personas que han sido clave en que esta travesía llegara a buen puerto:

Toda la gente que durante este tiempo ha pasado por el Departamento de Estadística e Investigación Operativa, profesores y compañeros de doctorado. Gracias por la formación, el apoyo y la ayuda cuando fue necesaria. Gracias por todos los buenos momentos que hicieron más agradable las horas de trabajo.

Mis directores Ángel Corberán y José María Sanchis, a quienes pertenece el mérito de, al menos, dos terceras partes de esta tesis. Gracias por todas las horas de trabajo. Gracias por vuestra amistad.

Mis padres y mis hermanas, porque sin ellos no sería quien soy ahora ni hubiera llegado hasta aquí. Gracias por estar siempre ahí. Gracias por toda una vida.

Y, finalmente, Fani. Compañera primero, amiga después y, ahora, esposa. Esta aventura del doctorado nos ha llevado mucho más lejos de lo que jamás habríamos imaginado y nos ha dado mucho más de lo que nunca hubiéramos soñado. Gracias por tu ayuda, por tu paciencia, por tu comprensión, por tu ánimo,... Simplemente, gracias.

Índice general

| | |
|--|-----------|
| Introducción | 1 |
| 1. Preliminares Matemáticos | 7 |
| 1.1. Teoría de Grafos | 7 |
| 1.2. Álgebra Lineal | 10 |
| 1.3. Teoría de Poliedros | 12 |
| 1.4. Teoría de la Complejidad Algorítmica | 13 |
| 1.5. Programación Lineal y Entera | 16 |
| 1.6. Combinatoria Poliédrica | 18 |
| 2. Algunos Problemas de Rutas | 21 |
| 2.1. Problemas de Rutas por Vértices o Arcos | 21 |
| 2.2. El Problema del Cartero Chino | 24 |
| 2.2.1. El Problema del Cartero Chino en un Grafo No Dirigido (CPP) | 24 |
| 2.2.2. El Prob. del Cartero Chino en un Grafo Dirigido (DCPP) | 25 |
| 2.2.3. El Problema del Cartero Chino en un Grafo Mixto (MCP) | 26 |
| 2.2.4. El Problema del Cartero Chino Con Viento (WPP) | 27 |
| 2.3. El Problema del Cartero Rural | 28 |
| 2.3.1. El Problema del Cartero Rural en un Grafo No Dirigido (RPP) | 28 |
| 2.3.2. El Prob. del Cartero Rural en un Grafo Dirigido (DRPP) | 29 |
| 2.3.3. El Problema del Cartero Rural en un Grafo Mixto (MRPP) | 30 |
| 2.4. El Problema del Cartero Rural con Viento (WRPP) | 31 |
| 2.5. El Problema General de Rutas | 32 |
| 2.5.1. El Problema General de Rutas sobre un Grafo No Dirigido (GRP) | 32 |
| 2.5.2. El Problema General de Rutas en un Grafo Mixto (MGRP) | 33 |
| 3. El Problema General de Rutas con Viento | 35 |
| 3.1. Descripción del problema | 35 |
| 3.1.1. Definición | 36 |
| 3.1.2. Casos particulares | 36 |
| 3.1.3. Aplicaciones | 37 |

| | |
|---|------------|
| 3.1.4. Soluciones del problema | 40 |
| 3.2. Formulación | 41 |
| 3.2.1. Simplificación | 41 |
| 3.2.2. Notación | 41 |
| 3.2.3. Formulación Lineal Entera | 42 |
| 3.3. El poliedro de soluciones | 43 |
| 3.3.1. Dimensión del poliedro | 43 |
| 3.4. Propiedades generales del poliedro | 44 |
| 3.4.1. Desigualdades de configuración | 44 |
| 3.4.2. Desigualdades de configuración débil | 50 |
| 4. Descripción del poliedro I | 53 |
| 4.1. Desigualdades triviales y de obligatoriedad | 53 |
| 4.2. Desigualdades de conectividad | 55 |
| 4.3. Desigualdades de corte R-impares | 56 |
| 5. Descripción del poliedro II | 59 |
| 5.1. Desigualdades K-C estándar | 59 |
| 5.2. Desigualdades K-C ₀₂ | 63 |
| 5.3. Desigualdades Path-Bridge | 64 |
| 5.4. Desigualdades Path-Bridge ₀₂ | 73 |
| 5.5. Desigualdades Honeycomb | 77 |
| 5.6. Desigualdades Honeycomb ₀₂ | 85 |
| 6. Descripción del poliedro III | 95 |
| 6.1. Desigualdades Zigzag pares | 98 |
| 6.2. Desigualdades Zigzag impares | 102 |
| 6.3. Relación de las nuevas facetas con otros problemas similares | 109 |
| 6.3.1. Las desigualdades Zigzag en el WPP | 110 |
| 6.3.2. Las desigualdades Zigzag en grafos no dirigidos | 111 |
| 6.3.3. Las desigualdades Zigzag en grafos dirigidos | 111 |
| 6.3.4. Las desigualdades Zigzag en grafos mixtos | 112 |
| 7. Branch And Cut | 123 |
| 7.1. Problema lineal inicial | 123 |
| 7.2. Cota superior inicial | 124 |
| 7.3. Tailing-off | 125 |
| 7.4. Ramificación | 125 |
| 7.5. Variaciones menos densas de algunas restricciones | 127 |
| 7.5.1. Desigualdades R-impares | 127 |
| 7.5.2. Desigualdades K-C y K-C ₀₂ | 127 |
| 7.6. Cota superior artificial y reinicio | 129 |
| 7.7. Filtrado de restricciones de R-imparidad | 131 |
| 7.8. Algoritmos de separación | 132 |
| 7.8.1. Desigualdad de conectividad | 133 |
| 7.8.2. Cortaduras R-impares | 133 |

| | |
|--|------------|
| 7.8.3. Desigualdades K-C | 135 |
| 7.8.4. Desigualdades Honeycomb | 141 |
| 7.8.5. Desigualdades K-C ₀₂ y Honeycomb ₀₂ | 145 |
| 7.8.6. Desigualdades Path-Bridge | 145 |
| 7.8.7. Desigualdades Zigzag | 150 |
| 8. Resultados computacionales | 157 |
| 8.1. Instancias | 157 |
| 8.1.1. Instancias en grafos windy | 157 |
| 8.1.2. Instancias en grafos mixtos | 160 |
| 8.1.3. Instancias en grafos no dirigidos | 161 |
| 8.1.4. Características de las instancias | 162 |
| 8.2. Pruebas computacionales | 162 |
| 9. Conclusiones | 175 |

Índice de figuras

| | |
|---|----|
| 3.1. Aplicaciones. Estructuras a inspeccionar | 38 |
| 3.2. Aplicaciones. Robot teledirigido | 38 |
| 3.3. Representación del problema como WRPP | 39 |
| 3.4. Ejemplo de construcción de un grafo de configuración | 46 |
| 4.1. Grafo de configuración. Desigualdades de corte R-impar | 57 |
| 4.2. Ejemplo de WGRP-tour. Cortes R-impares | 58 |
| 5.1. Grafo de configuración asociado a las desigualdades K-C | 60 |
| 5.2. Tour de coste mínimo en el grafo de configuración K-C | 61 |
| 5.3. WGRP-tour 1 (K-C estándar) | 61 |
| 5.4. WGRP-tour 2 (K-C estándar) | 62 |
| 5.5. WGRP-tour 3 (K-C estándar) | 62 |
| 5.6. Grafo de configuración asociado a las desigualdades K-C ₀₂ | 64 |
| 5.7. Grafo de configuración. Desigualdades Path-Bridge | 66 |
| 5.8. Tour de coste mínimo en el grafo de configuración Path Bridge | 66 |
| 5.9. Tour x^1 (Path-Bridge) | 67 |
| 5.10. WGRP-tour 1 (Path-Bridge) | 68 |
| 5.11. WGRP-tour 2 (Path-Bridge) | 68 |
| 5.12. WGRP-tour 3 (Path-Bridge) | 69 |
| 5.13. WGRP-tour 4 (Path-Bridge) | 70 |
| 5.14. WGRP-tour 5 (Path-Bridge) | 70 |
| 5.15. Grafo de configuración asociado a las desigualdades Path-Bridge ₀₂ | 75 |
| 5.16. WGRP-tour 1 (Path-Bridge ₀₂) | 76 |
| 5.17. Grafo de configuración asociado a una desigualdad Honeycomb | 79 |
| 5.18. Árbol T asociado al grafo configuración de la figura 5.17 | 80 |
| 5.19. WGRP-tour 1 (Honeycomb) | 80 |
| 5.20. WGRP-tour 2 (Honeycomb) | 81 |
| 5.21. WGRP-tour 3 (Honeycomb) | 81 |
| 5.22. WGRP-tour 4 (Honeycomb) | 82 |
| 5.23. WGRP-tour 5 (Honeycomb) | 82 |
| 5.24. Grafo configuración asociado a las desigualdades Honeycomb ₀₂ | 86 |
| 5.25. Vector x^d (Honeycomb ₀₂) | 88 |
| 5.26. WGRP-tour 1 tipo 4b (Honeycomb ₀₂) | 89 |

| | |
|---|-----|
| 5.27. WGRP-tour 2 tipo 4b (Honeycomb ₀₂) | 89 |
| 5.28. WGRP-tour 3 tipo 4c (Honeycomb ₀₂) | 90 |
| 5.29. WGRP-tour 4 tipo 4c (Honeycomb ₀₂) | 90 |
| 5.30. WGRP-tour 5 tipo 4d (Honeycomb ₀₂) | 91 |
| | |
| 6.1. Instancia del WGRP con solución fraccionaria 1 | 96 |
| 6.2. Solución fraccionaria 1 | 96 |
| 6.3. Solución óptima 1 | 96 |
| 6.4. Instancia del WGRP con solución fraccionaria 2 | 97 |
| 6.5. Solución fraccionaria 2 | 97 |
| 6.6. Solución óptima 2 | 97 |
| 6.7. Grafo de configuración asociado a una desigualdad Zigzag par | 99 |
| 6.8. Vector x^d , caso par | 100 |
| 6.9. Componentes seleccionadas, caso par | 100 |
| 6.10. WGRP-tour 1 (Zigzag par) | 101 |
| 6.11. WGRP-tour 2 (Zigzag par) | 101 |
| 6.12. WGRP-tour 3 (Zigzag par) | 101 |
| 6.13. WGRP-tour 4 (Zigzag par) | 102 |
| 6.14. Partición del grafo, desigualdades Zigzag impares | 103 |
| 6.15. Conjuntos \mathcal{H} , \mathcal{D} y \mathcal{F} | 103 |
| 6.16. Grafo de configuración asociado a una desigualdad Zigzag impar | 104 |
| 6.17. Vector x^d , caso impar | 106 |
| 6.18. Componentes seleccionadas, caso impar | 106 |
| 6.19. WGRP-tours 1 (Zigzag impar) | 107 |
| 6.20. WGRP-tours 2 (Zigzag impar) | 107 |
| 6.21. WGRP-tours 3 (Zigzag impar) | 108 |
| 6.22. WGRP-tours 4 (Zigzag impar) | 108 |
| 6.23. WGRP-tours 5 (Zigzag impar) | 108 |
| 6.24. WGRP-tours 6 (Zigzag impar) | 110 |
| 6.25. 3-Wheel en forma de Zigzag | 111 |
| 6.26. Grafo de configuración para las Zigzag en el DRPP | 112 |
| 6.27. Instancia con Zigzag válida para F2 pero no F1 | 120 |
| 6.28. Solución fraccionaria (formulación F2) | 120 |
| 6.29. Desigualdad Zigzag válida para F2 | 121 |
| 6.30. Desigualdad Zigzag no válida para F1 | 121 |
| 6.31. Solución fraccionaria (formulación F1) | 121 |
| 6.32. Zigzag para F1 como combinación lineal | 122 |
| | |
| 7.1. Coeficientes modificados para K-C | 128 |
| 7.2. Coeficientes modificados para K-C ₀₂ | 129 |
| 7.3. Solución que viola una restricción de conectividad y grafo G^* | 134 |
| 7.4. Grafo comprimido y cortadura de peso ≤ 2 | 134 |
| 7.5. Solución que viola una restricción K-C y grafo G^* | 135 |
| 7.6. Elección de M_0 y M_K | 136 |
| 7.7. Elección de M_0 y M_K alternativa | 137 |
| 7.8. Compresión de M_0 , M_K y componentes R-conexas | 138 |

7.9. Árbol generador de peso máximo en el grafo comprimido 138

7.10. Compresión de vértices hasta formar un camino entre M_0 y M_K 139

7.11. Configuración K-C que sí da lugar a una restricción K-C violada 140

7.12. Configuración K-C que no da lugar a una restricción K-C violada 140

7.13. Compresión de vértices para obtener una desigualdad K-C violada 141

7.14. Solución que viola una HC y grafo G^* 142

7.15. Etiquetado de los vértices de V_r 142

7.16. Conjuntos B_1, \dots, B_L 143

7.17. Árbol generador y compresión de vértices con grado 1 144

7.18. Configuración HC con $L=4$, $K=9$, que da lugar a una desigualdad
violada por x^* 144

7.19. Descomposición en bloques 145

7.20. Solución x^* que viola una desigualdad PB y grafo G^* 146

7.21. Elección de caminos candidatos y etiquetado de aristas débiles . 147

7.22. Candidatos para inicio de caminos 148

7.23. Configuración 2-PB asociada a una de las componentes candidatas 148

7.24. Elección de A , V_i^1 y V_i^2 149

7.25. Grafo comprimido G_s 149

7.26. Árbol en G_s y compresión de vértices 150

7.27. Configuración PB regular asociada a una de las componentes can-
didatas 151

7.28. Grafo ejemplo y solución fraccionaria para la separación de Z . . 151

7.29. Obtención de la cortadura que contenga el zigzag 152

7.30. Cortadura con zigzag 153

7.31. Cortadura sin zigzag 153

7.32. Numeración de los vértices del zigzag 154

7.33. Partición de los conjuntos V_1 y V_2 154

7.34. Partición final del grafo 155

7.35. Coeficientes de la desigualdad Zigzag identificada 155

Introducción

El origen de los Problemas de Rutas por Arcos se atribuye al problema de los puentes de Königsberg, resuelto por Euler en el siglo XVIII. Las diferentes partes en las que el río Pregel dividía la ciudad de Königsberg estaban comunicadas por siete puentes. La cuestión que resolvió Euler fue si era posible cruzar los siete puentes de la ciudad recorriendo exactamente una vez cada uno de ellos y volver al punto de partida. Este problema puede plantearse como un Problema de Rutas en un grafo, donde las diferentes partes de la ciudad están representadas por vértices y los puentes por aristas.

En general, un Problema de Rutas por Arcos consiste en, dado un grafo, diseñar una ruta recorriendo sus aristas que cumpla una serie de condiciones. El interés por el estudio de este tipo de problema estriba en que modeliza un gran número de problemas reales: reparto de correo, recogida de basuras, transporte público, mantenimiento de redes, etc. Los Problemas de Rutas por Arcos pueden ser formulados como Problemas de Optimización Combinatoria, donde el objetivo es hallar, de entre un conjunto numerable de soluciones posibles, aquella que minimiza o maximiza una función de coste.

Desgraciadamente, salvo para algunos de los problemas más sencillos, no es razonable esperar encontrar algoritmos capaces de resolver cualquier instancia de un problema con un número de operaciones que crezca polinómicamente con el tamaño de la instancia. Estos problemas para los que no parece que existan algoritmos de tiempo polinómico son los llamados NP-duros, y entre ellos se encuentra el problema tratado en esta tesis.

Entre los Problemas de Rutas NP-duros más conocidos y estudiados se encuentra el Problema del Viajante (TSP), en el cual, dado un grafo y unos costes asociados a las aristas, el objetivo es hallar una ruta de coste mínimo que pase exactamente una vez por cada vértice del grafo. Éste es caso particular de los Problemas de Rutas por vértices, aquéllos en los que la ruta ha de visitar algunos (o todos) de los vértices del grafo. Por otra parte, en los Problemas de Rutas por Arcos, el objetivo es diseñar una ruta que pase por algunas (o todas) de las aristas (y/o arcos) de un grafo. Algunos de los Problemas de Rutas por Arcos

más conocidos son el Problema del Cartero Chino (CPP), propuesto por Guan en 1962, en el que se busca una ruta de coste mínimo que pase al menos una vez por todas las aristas; o el Problema del Cartero Rural (RPP), presentado por Orloff (1974), similar al CPP pero donde no estamos obligados a recorrer todas las aristas del grafo, sino un subconjunto de ellas, llamadas aristas requeridas. El Problema General de Rutas (GRP), Orloff (1974), combina la demanda en las aristas con la demanda en los vértices. Así, una solución del GRP es una ruta que recorra todas las aristas requeridas y visite todos los nodos requeridos del grafo.

Además de por el tipo de demanda, los Problemas de Rutas pueden clasificarse también según el tipo del grafo sobre el que están definidos. Si los enlaces del grafo pueden ser recorridos en ambos sentidos, estamos ante un grafo dirigido y los enlaces reciben el nombre de aristas. En el caso en que los enlaces tienen una única dirección posible de recorrido, reciben el nombre de arcos, y el grafo se dice que es dirigido. Si existen tanto aristas como arcos en el grafo, estamos ante un grafo mixto. Por último, existe un cuarto tipo de grafos que contienen únicamente aristas, pero cada una de estas aristas tiene dos costes asociados, uno para cada sentido de recorrido. Este tipo de grafos recibe el nombre de *windy*, con viento, ya que representan la situación en que el cartero o vehículo encargado de recorrer la ruta tiene que avanzar a favor o en contra del viento, lo que le supone un menor o mayor coste. Desde un punto de vista más práctico, si los costes de una arista representan, por ejemplo, el tiempo que se tarda en recorrer una calle, estos grafos pueden representar situaciones en las que el tiempo de recorrido no es simétrico (debido, por ejemplo, a la densidad del tráfico). Además, los grafos *windy* generalizan a los grafos dirigidos, no dirigidos y mixtos.

Existen otros tipos de generalizaciones de los Problemas de Rutas que incorporan otra serie de características, como son la utilización de más de un vehículo para realizar las rutas, condiciones de ventanas temporales para el servicio de la demanda, etc. Este tipo de generalizaciones, sin embargo, se alejan del problema central de esta memoria, por lo que no entraremos en más detalles acerca de ellas.

El problema estudiado en esta tesis es el Problema General de Rutas con Viento (WGRP). En él, tenemos un grafo $G = (V, E)$ con un conjunto de vértices V , entre los cuales tenemos un subconjunto de vértices requeridos $V_R \subseteq V$, y un conjunto de aristas E , con un subconjunto de aristas requeridas $E_R \subseteq E$. Cada arista entre dos nodos i y j , que representamos por (i, j) , tiene dos costes asociados: c_{ij} es el coste de recorrerla de i a j , y c_{ji} es el coste de recorrerla de j a i . El objetivo es hallar una ruta de coste mínimo que recorra al menos una vez todas las aristas requeridas y visite todos los vértices requeridos.

Este problema tiene un gran interés teórico, puesto que generaliza la mayor parte de los Problemas de Rutas por Arcos con un único vehículo. Por ejemplo,

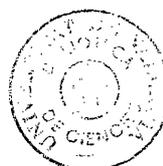
en el caso de que todas las aristas sean requeridas, tenemos el WPP, y si no hay aristas requeridas y todos los vértices son requeridos, tenemos el GTSP. Así pues, los resultados teóricos obtenidos en el estudio de este problema podrán ser de aplicación a muchos otros. También los métodos de resolución, puesto que los algoritmos desarrollados para el WGRP pueden ser adaptados fácilmente para resolver instancias de sus casos particulares.

En cuanto a sus aplicaciones prácticas, en esta tesis presentamos una situación real que puede ser modelizada como un WGRP. Se trata del diseño de rutas para robots escaladores encargados de la inspección de estructuras arquitectónicas complejas, como puentes o edificios. Estos robots son teledirigidos y poseen una cámara mediante la cual los operarios pueden observar las superficies de la estructura en busca de posibles daños. Las superficies a explorar serán representadas por aristas requeridas, mientras que los movimientos realizados por el robot para moverse de una superficie a otra corresponden a aristas no requeridas. La necesidad de dos costes para cada arista viene impuesta por el efecto de la gravedad en el consumo de energía dependiendo del sentido de movimiento del robot.

En esta tesis abordamos la resolución del WGRP desde el punto de vista de la Combinatoria Poliédrica, uno de los enfoques más utilizados en los problemas NP-duros. De esta manera, tratamos de resolver el problema como un Problema Lineal, para lo cual es necesario conocer una descripción lo suficientemente completa del poliedro definido por la envoltura convexa del espacio de soluciones posibles.

En efecto, si conociéramos todas las desigualdades que definen faceta del poliedro, tendríamos su descripción completa y podríamos hallar la solución óptima del problema resolviendo un Problema Lineal. En la práctica, y puesto que se trata de un problema NP-duro, esto no es posible, por lo que hemos de conformarnos con una descripción parcial. En esta memoria, presentamos una serie de familias de desigualdades válidas para el WGRP que definen facetas del poliedro. Las siguientes clases de desigualdades, propuestas anteriormente para otros problemas, han sido generalizadas al WGRP en este trabajo, en el que también se ha demostrado que inducen faceta de su poliedro asociado:

- Desigualdades triviales.
- Desigualdades de obligatoriedad.
- Desigualdades de conectividad.
- Desigualdades de cortes R-impares.
- Desigualdades K-C y K-C₀₂.
- Desigualdades Path-Bridge y Path-Bridge₀₂.



- Desigualdades Honeycomb y Honeycomb₀₂.

Estas desigualdades, a excepción de las triviales y las de obligatoriedad, tienen una serie de características comunes y se denominan *desigualdades de configuración*. Así, todas ellas pueden asociarse a un grafo que resulta de dividir el grafo original en subconjuntos de vértices y comprimir las aristas no requeridas paralelas (entre los mismos pares de subconjuntos) en una sola. Entonces, los coeficientes de las variables asociadas a las aristas entre vértices pertenecientes a un mismo subconjunto serán 0, mientras que los coeficientes de las aristas requeridas en paralelo serán iguales. El grafo así comprimido se denomina *grafo de configuración*. Hemos demostrado que, si una desigualdad induce una faceta del WGRP definido sobre el grafo de configuración, también define una faceta del poliedro correspondiente al problema en el grafo original. Sin embargo, a diferencia de lo que ocurre para otros casos particulares del WGRP, la afirmación recíproca no es cierta. Es decir, existen desigualdades que definen facetas del poliedro del WGRP, distintas de las triviales y las de obligatoriedad, que no son desigualdades de configuración. En esta tesis, presentamos una nueva familia de desigualdades, a las que llamamos *desigualdades Zigzag* que definen faceta del poliedro del WGRP sin ser desigualdades de configuración. Al igual que las otras, estas desigualdades pueden definirse sobre un grafo de configuración, donde las variables asociadas a las aristas “internas” de los subconjuntos tendrán coeficiente 0 en la desigualdad. Sin embargo, existen aristas paralelas en el grafo configuración que tienen coeficientes distintos en la desigualdad. Hasta donde sabemos, este es el primer caso de desigualdades para problemas de Rutas por Arcos con esta característica, por lo que hemos introducido el concepto de *desigualdades de configuración débil*.

Puesto que es imposible incorporar explícitamente a la formulación la descripción parcial de la envoltura convexa del espacio de soluciones, esta puede ir siendo añadida de manera iterativa en lo que se conoce como un algoritmo de Planos de Corte. En este tipo de algoritmos, partimos de un Problema Lineal que es una versión relajada del problema a resolver. Este problema puede ser resuelto fácilmente, pero la solución obtenida no será, generalmente, una solución válida de nuestro problema, es decir, estará fuera de la envoltura convexa del espacio de soluciones. Existirá, por tanto, una desigualdad válida para el problema que no es satisfecha por la solución, es decir, *violada por la solución*. Podemos, entonces, intentar buscar, entre las familias conocidas, una desigualdad violada que nos permita “cortar” la solución no válida. Para ello se emplean los llamados *algoritmos de separación*, que se encargan de encontrar, si existe, dicha desigualdad. Sin embargo, a menudo, el problema de, dada una solución no válida, determinar si dentro de una determinada familia de desigualdades existe alguna violada, es NP-duro en sí mismo, por lo que deberemos recurrir a algoritmos heurísticos. Si mediante los algoritmos de separación hemos podido identificar alguna desigualdad violada, podemos añadirla al Problema Lineal actual y volver a resolverlo. Este proceso se repite hasta que, o bien obtenemos

una solución factible, que será óptima, o bien no somos capaces de hallar más desigualdades violadas.

Otra alternativa a los algoritmos de Planos de Corte son los algoritmos de Ramificación y Acotación. Estos algoritmos se basan en la construcción de un árbol de enumeración para explorar el espacio de soluciones, utilizando cotas inferiores obtenidas mediante la resolución de Problemas Lineales así como cotas superiores pertenecientes a soluciones válidas conocidas, para limitar dicha búsqueda.

Si en cada nodo del árbol de enumeración ejecutamos un algoritmo de Planos de Corte sobre el Problema Lineal correspondiente, podemos mejorar las cotas obtenidas más rápidamente. Estos algoritmos se conocen como algoritmos de Ramificación y Corte (Branch and Cut), y fueron aplicados por primera vez por Padberg y Rinaldi (1990). En esta memoria presentamos un algoritmo de Ramificación y Corte para el WGRP. Para probar la eficiencia de dicho algoritmo, hemos utilizado un gran número de instancias, tanto del WGRP como de los problemas que aparecen como casos particulares del WGRP. Algunas de estas instancias han sido generadas expresamente para este trabajo, mientras que otras son instancias tomadas de la literatura conocida. Los tamaños de las instancias varían desde los 7 nodos y 10 enlaces en los grafos de las instancias más pequeñas, hasta los 1000 nodos y más de 4000 enlaces para las más grandes. Para terminar la memoria, presentamos las conclusiones derivadas de este trabajo y las líneas de trabajo a seguir en el futuro.

Capítulo 1

Preliminares Matemáticos

Este capítulo recoge algunos conceptos básicos de Teoría de Grafos, Álgebra Lineal, Programación Lineal, Teoría de Poliedros y Teoría de la Complejidad Algorítmica que son utilizados en capítulos posteriores.

1.1. Teoría de Grafos

Muchos problemas que encontramos en el mundo real, entre ellos los Problemas de Rutas de Vehículos, pueden ser formulados mediante la Teoría de Grafos. Algunos problemas que a priori parecen complicados y ambiguos, representados en términos de Teoría de Grafos son claros y precisos, permitiendo así un acercamiento matemático más adecuado para su resolución. En esta sección, introduciremos algunos conceptos y resultados de Teoría de Grafos que hemos empleado en el presente trabajo. Algunas referencias clásicas en esta área son Harady (1969), Berge (1973), Bondy y Murty (1976) y Christofides (1975).

Sea V un conjunto finito no vacío y sea E una familia finita de pares no ordenados (no necesariamente distintos) (i, j) de elementos de V (no necesariamente distintos). Llamaremos *grafo no dirigido* al par $G = (V, E)$, *vértices* de G a los elementos de V y *aristas* de G a los elementos de E .

Si el par $e = (i, j)$ es una arista de G , diremos que i y j son los *vértices extremos* de e , o que la arista e une los vértices i y j , o que la arista e es incidente con los vértices i y j , o que los vértices i y j son adyacentes.

A las aristas de la forma (i, i) las llamaremos *bucles*. Si dos aristas tienen

los mismos vértices extremos diremos que son paralelas. Diremos que un grafo es *simple* si no tiene bucles ni aristas en paralelo y diremos que un grafo es *completo* si es simple y para cada par de vértices $i \neq j$ existe una arista (i, j) entre ellos.

A veces, por simplicidad, denotaremos la arista (i, j) simplemente por ij , y usaremos una coma entre los dos vértices si es necesario (por ejemplo para la arista $i, i + 1$).

Dado un vértice i de un grafo G , denotaremos por $\delta_G(i)$ al conjunto de todas las aristas de G incidentes con i , o simplemente $\delta(i)$ si no hay ambigüedad. Llamaremos *grado* del vértice i a la cardinalidad de $\delta(i)$, i.e., el número de aristas incidentes con el vértice i , y diremos que un vértice i es *par* (resp. *impar*) si su grado es par (resp. impar). Diremos que G es un grafo par si todos sus vértices son pares.

Teorema 1.1.1 *Sea G un grafo no dirigido. G contiene un número par de vértices impares.*

Si S_1 y S_2 son dos subconjuntos disjuntos de V , denotaremos por $(S_1 : S_2)$ al conjunto de todas las aristas de G con un vértice extremo en S_1 y el otro en S_2 . En particular, si representamos por \bar{S} a $V \setminus S$, llamaremos *cortadura* de G a $(S : \bar{S})$ para cualquier $S \subseteq V$ no vacío, y diremos que una cortadura es par (resp. impar) si el número de aristas que contiene es par (resp. impar). En particular, $\delta(i)$, $i \in V$, es una cortadura de G .

Teorema 1.1.2 *$(S : \bar{S})$ es una cortadura impar de G si y sólo si S y \bar{S} contienen un número impar de vértices impares.*

Si $G = (V, E)$ y $G_1 = (V_1, E_1)$ son dos grafos tales que $V_1 \subseteq V$ y $E_1 \subseteq E$, diremos que G_1 es un subgrafo de G . Dado un subconjunto $V' \subseteq V$ denotaremos por $E(V')$ al conjunto de aristas de G con ambos extremos en V' y llamaremos *subgrafo* de G inducido por V' al grafo $G(V') = (V', E(V'))$. Del mismo modo, dado un subconjunto E' de E , denotaremos por $V(E')$ al conjunto de vértices de G que son incidentes con al menos una arista de E' , y llamaremos subgrafo de G inducido por E' al grafo $G(E') = (V(E'), E')$.

Sea $G = (V, E)$ un grafo y sean $u, v \in V$. Llamaremos *camino* entre u y v a cualquier secuencia $w = (i_0 i_1, i_1 i_2, \dots, i_{k-1} i_k)$ de aristas de G con $i_0 = u$ y $i_k = v$, y diremos que w contiene o recorre las aristas $i_0 i_1, i_1 i_2, \dots, i_{k-1} i_k$, así como que contiene o pasa por los vértices $i_0, i_1, i_2, \dots, i_{k-1}, i_k$. Si no hay ambigüedad, también representaremos w como una secuencia de vértices $(i_0, i_1, i_2, \dots, i_{k-1}, i_k)$ o como el conjunto de aristas $i_j i_{j+1}, 0 \leq j \leq k - 1$.

Si w es un camino y la arista $i_k i_0 \in E$, llamaremos *tour* o camino cerrado a la secuencia $(i_0 i_1, i_1 i_2, \dots, i_{k-1} i_k, i_k i_0)$. Llamaremos *ciclo de longitud k* a cualquier tour $(i_0 i_1, i_1 i_2, \dots, i_{k-1} i_k, i_k i_0)$ tal que $i_s \neq i_t$ $0 \leq s, t \leq k$ (no pasa más de una vez por el mismo vértice). Un ciclo *hamiltoniano* en el grafo G es un ciclo que pasa por cada vértice de G exactamente una vez.

Podemos representar un grafo no dirigido en el plano mediante un diagrama de puntos y líneas. Cada punto corresponde a un vértice del grafo y para cada arista del grafo entre dos vértices u y v hay una línea que une los dos puntos correspondientes.

Teorema 1.1.3 *Si w es un tour sobre las aristas de G , entonces w pasa un número par de veces (quizás cero) por las aristas de cualquier cortadura de G .*

Diremos que un grafo G es *conexo* si entre cada par de vértices de G existe al menos un camino. Por convenio, un grafo con un único vértice es conexo.

Una *componente conexa* de G es un subgrafo conexo de G que es inducido por un conjunto de vértices de G y es maximal respecto a la cardinalidad del subconjunto de vértices. Es inmediato comprobar que un grafo G es conexo si y sólo si tiene una única componente conexa. Diremos que una arista e es un *punte* de G si el subgrafo de G inducido por $E \setminus \{e\}$ tiene más componentes conexas que G .

Un tour *Euleriano* en el grafo G es un tour que recorre cada arista de G exactamente una vez y se dice que un grafo es Euleriano si existe un tour Euleriano en él.

Teorema 1.1.4 (Euler) *G es Euleriano si y sólo si es conexo y par.*

Sea $G = (V, E)$ un grafo no dirigido y sea $c : E \rightarrow R$ una función que asigna un coste o longitud c_e a cada arista $e \in E$. Sea $F \subseteq E$. Llamaremos *coste* o longitud de F a $c(F) = \sum_{f \in F} c_f$. Del mismo modo, si $w = (i_0 i_1, i_1 i_2, \dots, i_{k-1} i_k)$

es un camino en G , llamaremos coste o longitud de w a $c(w) = \sum_{j=0}^{k-1} c_{i_j i_{j+1}}$

Sea V un conjunto finito no vacío y sea A una familia finita de pares ordenados (i, j) (no necesariamente distintos) de elementos de V ordenados (no necesariamente distintos). Llamaremos *grafo dirigido* al par $G = (V, A)$, vértices de G a los elementos de V y *arcos* de G a los elementos de A .

Sea V un conjunto finito no vacío, E una familia finita de pares no ordenados (no necesariamente distintos) (i, j) de elementos de V (no necesariamente

distintos) y sea A una familia finita de pares ordenados (i, j) (no necesariamente distintos) de elementos de V (no necesariamente distintos). Llamaremos *grafo mixto* a la terna $G = (V, E, A)$, vértices de G a los elementos de V , aristas de G a los elementos de E , arcos de G a los elementos de A y, en general, *enlaces*, a los arcos y aristas indistintamente.

Del mismo modo que un grafo no dirigido, un grafo dirigido puede representarse en el plano mediante un diagrama de puntos y flechas y un grafo mixto puede representarse mediante un diagrama de puntos, líneas y flechas. Todas las definiciones y resultados que hemos visto para grafos no dirigidos pueden adaptarse a grafos dirigidos y grafos mixtos.

Si $a = (i, j) \in A$ es un arco de G , diremos que i y j son los vértices extremos de a , i es el vértice origen y j el vértice final, o que el arco a sale de i y entra en j , o que el arco a une los vértices i y j , o que el arco a es incidente con los vértices i y j , o que los vértices i y j son adyacentes.

Si dos arcos tienen el mismo vértice origen y el mismo vértice final, diremos que son paralelos.

Dado un vértice i de un grafo mixto G , denotaremos por $\delta_G(i)$ al conjunto de todos los enlaces (arcos y aristas) de G incidentes con i , o simplemente $\delta(i)$ si no hay ambigüedad. Llamaremos grado del vértice i a la cardinalidad de $\delta(i)$, que es la suma del número de aristas incidentes con i , el número de arcos que salen de i y el número de arcos que entran en i . Y diremos que un vértice i es par (resp. impar) si su grado es par (resp. impar). Diremos que G es un grafo par si todos sus vértices son pares.

Llamaremos grado de entrada (salida) de un vértice i , al número de arcos que entran en (salen de) i . Un grafo en el que para todo vértice, su grado de entrada y de salida coinciden, recibe el nombre de grafo simétrico.

Diremos que un grafo mixto $G = (V, E, A)$ es débilmente conexo si entre cada par de vértices $i, j \in V$ existe al menos una secuencia de enlaces (sin tener en cuenta su sentido, en el caso de arcos) uniendo i y j .

Diremos que G es fuertemente conexo si entre cada par de vértices $i, j \in V$ existe al menos un camino de i a j y otro de j a i .

1.2. Álgebra Lineal

Denotaremos por R^n el espacio de todos los vectores columna de n componentes reales. Sean $x_i \in R^n$, $i = 1, \dots, m$. Diremos que un vector $y \in R^n$ es una

combinación lineal de los x_i si y sólo si y se puede expresar como $y = \sum_{i=1}^m \lambda_i x_i$, con $\lambda_i \in R$ cualesquiera.

Si, además, $\sum_{i=1}^m \lambda_i = 1$, diremos que y es una combinación afín de los vectores x_i . Una combinación afín que cumpla $\lambda_i \geq 0$ para cada i es una combinación convexa y una combinación lineal que cumpla $\lambda_i \geq 0$ para cada i es una combinación cónica.

Dado un conjunto de vectores $S \subseteq R^n$, llamaremos envoltura lineal (resp. afín, convexa y cónica) de S al conjunto de todas las combinaciones lineales (resp. afines, convexas y cónicas) que se pueden construir con vectores de S , y la representaremos por $\text{lin}(S)$ (resp. $\text{afín}(S)$, $\text{convex}(S)$ y $\text{cone}(S)$). Diremos que S es un subespacio lineal (resp. subespacio afín, convexo y cono) de R^n si $S = \text{lin}(S)$ (resp. $\text{afín}(S)$, $\text{convex}(S)$, $\text{cone}(S)$).

Diremos que los vectores $x_i \in R^n$, $1 \leq i \leq m$, son linealmente independientes si ninguno de ellos es combinación lineal de los demás, o equivalentemente si $\sum_{i=0}^m \lambda_i x_i = 0$ implica que $\lambda_i = 0$ para todo i . En caso contrario diremos que son linealmente dependientes. Del mismo modo, diremos que los vectores $x_i \in R^n$, $1 \leq i \leq m$ son afinmente independientes si ninguno de ellos es combinación afín de los demás, o equivalentemente si $\sum_{i=0}^m \lambda_i x_i = 0$, con $\sum_{i=0}^m \lambda_i = 0$, implica que $\lambda_i = 0$ para todo i . En otro caso, diremos que los vectores son afinmente dependientes.

Teorema 1.2.1 Sea $S \subseteq R^n$. Son equivalentes:

- I) S es afinmente independiente.
- II) Dado $y \in S$, $\{x - y : x \in S, x \neq y\}$ es linealmente independiente.
- III) Dado $y \in R^n$, $\{x - y : x \in S\}$ es afinmente independiente.

Dado $S \subseteq R^n$, llamaremos rango de S al número máximo de vectores de S linealmente independientes, y lo denotaremos por $\text{rg}(S)$. Del mismo modo, llamaremos rango afín de S al número máximo de vectores de S afinmente independientes, y lo denotaremos por $\text{rgaf}(S)$. Llamaremos rango de una matriz A , $\text{rg}(A)$, al rango del conjunto de sus vectores columna, que es el mismo que el rango de sus vectores fila.

Teorema 1.2.2 Sea $S \subseteq R^n$

- I) Si $0 \in \text{afín}(S)$, entonces $\text{rgaf}(S) = \text{rg}(S) + 1$.
 II) Si $0 \notin \text{afín}(S)$, entonces $\text{rgaf}(S) = \text{rg}(S)$

Si S es un subespacio lineal de R^n , llamaremos base de S a cualquier subconjunto finito B de vectores linealmente independientes de S tal que $\text{lin}(B) = S$. Todas las bases de un mismo subespacio lineal tienen el mismo número de vectores, y a ese número lo llamaremos dimensión de S . Si S es un subespacio afín de R^n , existe un único subespacio lineal S' de R^n tal que $S' = \{x - x^* : x \in S\}$, para cualquier $x^* \in S$. Llamaremos dimensión de S a la dimensión de S' . Por último, si S es un subconjunto arbitrario de R^n , llamaremos dimensión de S a la dimensión de $\text{afín}(S)$ y la denotaremos por $\text{dim}(S)$.

1.3. Teoría de Poliedros

Sean $a \in R^n$ y $\alpha \in R$. Llamaremos hiperplano de R^n al conjunto $\{x \in R^n : a^T x = \alpha\}$ y llamaremos semiespacio de R^n al conjunto $\{x \in R^n : a^T x \leq \alpha\}$. Llamaremos poliedro de R^n a la intersección de un número finito de semiespacios de R^n o, equivalentemente, al conjunto de soluciones de un sistema de inecuaciones de la forma $Ax \leq b$, donde A es una matriz $m \times n$ y $b \in R^m$. Llamaremos polítopo a un poliedro acotado. Diremos que un poliedro P es de dimensión completa si $\text{dim}(P) = n$.

Teorema 1.3.1 $P \subseteq R^n$ es un poliedro si y sólo si existen V y E , subconjuntos finitos de R^n , tales que $P = \text{convex}(V) + \text{cone}(E)$.

Sean $a \in R^n$, $\alpha \in R$. Diremos que $a^T x \leq \alpha$ es una desigualdad válida para el poliedro P si $P \subseteq \{x \in R^n : a^T x \leq \alpha\}$. Diremos que $F \subseteq R^n$ es una cara del poliedro P si existe una desigualdad válida $a^T x \leq \alpha$ para P tal que $F = P \cap \{x \in R^n : a^T x = \alpha\}$. En este caso diremos que la cara F está inducida por la desigualdad $a^T x \leq \alpha$. Varias desigualdades pueden inducir la misma cara de un poliedro P . En este caso diremos que esas desigualdades son equivalentes respecto a P . Obviamente, un poliedro P es una cara de sí mismo. Llamaremos caras propias a las demás caras no vacías de un poliedro P . Nótese que, con esta definición, los vértices y aristas de un poliedro son caras. Así mismo, llamaremos faceta del poliedro P a cualquier cara propia no vacía de P que sea maximal respecto a la inclusión de conjuntos. De esta manera, el concepto de faceta se corresponde con la idea coloquial de cara de un poliedro.

Teorema 1.3.2 Sea $P \subseteq R^n$ un poliedro y sea F una cara propia no vacía de P inducida por una desigualdad válida $a^T x \leq \alpha$. Entonces, son equivalentes:

- i) F es una faceta de P .
- ii) $\dim(F) = \dim(P) - 1$.
- iii) Si $b^T x \leq \beta$ es válida para P , $F \subseteq P \cap \{x \in R^n : b^T x = \beta\}$ y $\text{afín}(P) = \{x \in R^n : Ax = d\}$, donde A es una matriz $m \times n$, $d \in R$, entonces existen $\lambda \in R^m$ y $\mu \geq 0$ tales que $b^T = \mu a^T + \lambda^T A$.

Llamaremos vértice del poliedro P a cualquier cara propia no vacía de P que sea minimal respecto a la inclusión de conjuntos, es decir, a cualquier cara de P que esté formada por un único punto $F = \{v\}$.

Teorema 1.3.3 $v \in P$ es un vértice si y sólo si v no puede expresarse como combinación convexa de otros puntos de P .

Llamaremos poliedro entero a cualquier poliedro cuyos vértices tienen todas sus componentes enteras.

Los artículos de Bachem y Grötschel (1982) y Pulleyblank (1983) y los libros de Schrijver (1986) y Nemhauser y Wolsey (1988) proporcionan una información abundante sobre los conceptos y resultados de la Teoría de Poliedros y la Combinatoria Poliédrica.

1.4. Teoría de la Complejidad Algorítmica

En Matemáticas, la palabra problema generalmente hace referencia a una cuestión concreta a resolver o una tarea a realizar. Sin embargo, nosotros llamaremos problema a una cuestión general planteada en términos de algunos "parámetros abiertos" para la cual se busca una "solución". Un problema se describe mediante una lista de sus "parámetros abiertos" y una lista de las propiedades que deben satisfacer sus soluciones. Si especificamos todos los "parámetros abiertos" con unos valores concretos, entonces tenemos lo que llamaremos instancia del problema. Por ejemplo, el problema "resolver un sistema de ecuaciones lineales" puede ser definido como "dada una matriz A $m \times n$ y un vector $b \in R^m$, ¿existe algún vector $x \in R^n$ tal que $Ax = b$?" A y b son los parámetros abiertos. Una instancia de este problema sería, por ejemplo, encontrar x, y, z tales que:

$$\begin{aligned} 3x - y + 2z &= 1 \\ -x + 7y - 4z &= 3 \end{aligned}$$

Llamaremos algoritmo a una secuencia de pasos sin ambigüedad que llevan a la solución de un problema. Diremos que un algoritmo resuelve un problema

si encuentra una solución para cada instancia del problema. Para determinar la “eficiencia” de un algoritmo debemos considerar todos los “recursos” que emplea dicho algoritmo. Por ejemplo, si utilizamos un ordenador para ejecutar un algoritmo, el uso de los distintos recursos del ordenador (tiempo de ejecución, memoria, etc.) nos proporcionará distintas medidas para evaluar la eficiencia del algoritmo. Sin embargo, normalmente el tiempo de ejecución es el recurso dominante y se acepta que el algoritmo más rápido para resolver un problema es el más eficiente. Por último, para obviar las diferencias que hay entre los distintos ordenadores, podemos definir el tiempo de ejecución de un algoritmo como el número de operaciones elementales requeridas para llegar a una solución del problema.

Naturalmente, el tiempo de ejecución depende del “tamaño” de la instancia, luego es razonable expresar el tiempo de ejecución de un algoritmo en función del tamaño de las instancias. Vamos a tratar todo esto de una manera más precisa.

Codificar una instancia significa representarla mediante una hilera finita de símbolos (que servirá de entrada al ordenador). Llamaremos tamaño o longitud de una instancia I al número de símbolos necesarios para codificarla, y lo denotaremos por $l(I)$. Claramente, $l(I)$ depende no sólo de I sino del esquema de codificación empleado. Asumiremos que hemos utilizado un esquema de codificación conciso (no emplea símbolos innecesarios) y que los números son codificados en una base fija.

Sea Π un problema, A un algoritmo que resuelve Π y sea I una instancia de Π . Llamaremos tiempo de ejecución de A para resolver I , $t_A(I)$, al número de pasos elementales necesarios para llegar a una solución de I . Llamaremos función de tiempo de ejecución de A a la función $f_A : \mathbb{N} \rightarrow \mathbb{N}$ tal que $f_A(n) = \max\{t_A(I) : l(I) \leq n\}$, es decir, que asigna a cada $n \in \mathbb{N}$ el máximo tiempo de ejecución necesario para resolver cualquier instancia I de tamaño menor o igual que n . Diremos que A es un algoritmo polinómico si existe un polinomio $p(n)$ tal que $f_A(n) = O(p(n))$, es decir, si existe una constante M tal que $|f_A(n)| \leq M |p(n)|$, $\forall n$. En otro caso diremos que A es un algoritmo exponencial.

Aceptaremos que los algoritmos polinómicos son eficientes, puesto que su tiempo de ejecución crece de un modo “razonable” a medida que crece el tamaño de la instancia (aunque este crecimiento puede ser considerablemente grande si el grado del polinomio lo es). De este modo, podemos decir que un problema es “fácil” si puede ser resuelto mediante un algoritmo polinómico y que es “difícil” en otro caso. Para obtener una clasificación más rigurosa de problemas según su complejidad algorítmica procederemos como sigue:

Consideremos primero los problemas de decisión, es decir, problemas cuyas instancias tienen solamente dos posibles soluciones: “sí” o “no”. Denotaremos por P la clase de todos los problemas de decisión que pueden ser resueltos mediante

un algoritmo polinómico. Por ejemplo, el problema Π_1 : “dado un grafo G , ¿contiene un ciclo?” es un problema de decisión, y puede probarse que está en P . Sin embargo, para el problema de decisión Π_2 : “dado un grafo G , ¿contiene un ciclo hamiltoniano?”, no se conoce ningún algoritmo polinómico que lo resuelva, luego no se sabe si está en P o no. Observemos que una instancia I de Π_2 cuya solución sea “sí” contendrá un ciclo hamiltoniano y, conociendo ese ciclo, podemos comprobar la veracidad de la solución “sí” en un tiempo polinómico en $l(I)$ (basta comprobar que efectivamente es un ciclo hamiltoniano). Diremos que los problemas con esta propiedad están en la clase NP .

En general, diremos que un problema de decisión Π está en la clase NP si para cada instancia I de Π con solución “sí” existe una estructura S (el ciclo hamiltoniano en el problema Π_2) tal que, conociendo S , podemos comprobar la veracidad de la solución “sí” en un tiempo polinómico en $l(I)$. Notemos que en esta definición se exige solamente la existencia de S y no necesariamente un procedimiento para encontrarla.

Obviamente $P \subseteq NP$. La cuestión “¿es $P = NP$?” es el principal problema abierto de la Teoría de la Complejidad Algorítmica, aunque una conjetura generalmente aceptada es que la inclusión es estricta.

Sean Π_1 y Π_2 dos problemas de decisión. Llamaremos transformación polinómica de Π_1 en Π_2 a cualquier algoritmo polinómico que para cada instancia I_1 produce una instancia I_2 de Π_2 tal que la solución para I_1 es “sí” si y sólo si la solución para I_2 es “sí”. Diremos que un problema de decisión Π^* es NP -completo si está en NP y cualquier otro problema de NP es transformable polinómicamente en Π^* . Esta definición implica que si Π^* es NP -completo y $\Pi^* \in P$, entonces $P = NP$. Por lo tanto, si aceptamos la conjetura de que $P \neq NP$, no debe existir ningún algoritmo polinómico que resuelva Π^* . En este sentido, los problemas NP -completos son los más difíciles de la clase NP . De la definición de NP -completo se deduce que un problema Π es NP -completo si está en NP y otro problema NP -completo conocido es transformable polinómicamente en Π . Este hecho ha servido para demostrar que una gran cantidad de problemas son NP -completos. El primer problema NP -completo conocido fue el Problema de la Satisfabilidad (Cook, 1971). El Problema del Ciclo Hamiltoniano considerado anteriormente es también NP -completo.

Diremos que un problema Π_1 es reducible polinómicamente a otro problema Π_2 si:

1. Existe un algoritmo A_1 que resuelve Π_1 usando como subrutina un algoritmo A_2 para Π_2 .
2. A_1 es un algoritmo polinómico si y sólo si A_2 es un algoritmo polinómico.



Los problemas que vamos a tratar en esta tesis son problemas de Optimización Combinatoria, que no son propiamente problemas de decisión (aunque existen versiones de estos problemas como problemas de decisión y se puede demostrar que son equivalentes en su complejidad) y, por lo tanto, no pueden ser *NP-completos*. Así mismo, hay problemas de decisión en los que son transformables polinómicamente todos los problemas de *NP* pero no está demostrada su pertenencia a *NP* y, por lo tanto, tampoco se les puede clasificar como *NP-completos*. Para todos estos tipos de problemas se reserva el término *NP-duro* (*NP-hard*). Un problema es *NP-duro* si todos los problemas de *NP* son reducibles (no necesariamente transformables) polinómicamente a él.

Una gran cantidad de problemas de optimización son *NP-duros*, y, por lo tanto, si asumimos que $P \neq NP$ no podremos encontrar algoritmos polinómicos que los resuelvan. Por este motivo, la investigación actual sobre estos problemas se divide básicamente en tres ramas distintas pero complementarias:

- Diseñar algoritmos heurísticos que sean rápidos y que produzcan soluciones “próximas” a la solución óptima.
- Buscar casos especiales de esos problemas que sí puedan ser resueltos en tiempo polinómico.
- Diseñar algoritmos que busquen la solución óptima.

Aunque hemos visto que no podemos esperar encontrar algoritmos polinómicos que resuelvan cualquier instancia, es interesante diseñar algoritmos exponenciales exactos por varios motivos. Primero, para resolver instancias de tamaño no demasiado grande. Si el tiempo de ejecución crece de modo exponencial con el tamaño de la instancia, no podremos resolver instancias de “cualquier” tamaño, pero al menos sí podremos resolver instancias “pequeñas”, y cuanto mejor sea el algoritmo, mayores serán las instancias que seremos capaces de resolver. En segundo lugar, la definición de tiempo de ejecución se hace sobre el peor caso, es decir, para resolver una instancia de tamaño n necesitamos realizar $t_A(n)$ operaciones como máximo. Pero puede ocurrir que instancias concretas de gran tamaño tomadas del mundo real se resuelvan en un tiempo razonable.

Una presentación más extensa y rigurosa de estos conceptos puede encontrarse en Papadimitriou (1976, 1984).

1.5. Programación Lineal y Entera

Llamaremos problema de Programación Lineal, abreviadamente *PL*, al problema de minimizar (maximizar) una función lineal $f(x) = c^T x$, $c \in R^n$, que

llamaremos función objetivo, sobre un poliedro $P \subseteq R^n$. Llamaremos solución posible a cualquier $x \in P$ y llamaremos solución óptima a cualquier $x^* \in P$ tal que:

$$c^T x^* = \min\{c^T x : x \in P\}$$

Teorema 1.5.1 *Si P tiene al menos un vértice y $\min\{c^T x : x \in P\}$ es finito, entonces existe al menos un vértice en P que es solución óptima.*

Teorema 1.5.2 *Para cada vértice x^* de P , existe un vector $c \in R^n$ tal que $c^T x^* < c^T x$ para cada $x \in P \setminus \{x^*\}$.*

Corolario 1.5.3 *Un poliedro P es entero si y sólo si para cada $c \in R^n$ tal que $\min\{c^T x : x \in P\}$ es finito, existe un vector entero $x^* \in P$ tal que:*

$$c^T x^* = \min\{c^T x : x \in P\}$$

En 1945, Dantzig desarrolló un algoritmo llamado método simplex (Dantzig, 1963) para la resolución de Problemas de Programación Lineal. Para la demostración de que este algoritmo no es polinómico se encontró un conjunto de instancias del PL para las que el simplex necesita hacer un número de operaciones que depende exponencialmente del tamaño de la instancia. Khachian (1979) desarrolló un algoritmo polinómico, llamado método de las elipsoides, para la resolución del PL. La aportación más importante de este algoritmo es la constatación de que el PL es un problema de la clase P.

Llamaremos problema de Programación Lineal Entera o, abreviadamente, PLE, al problema de minimizar (maximizar) una función lineal sobre los vectores enteros de un poliedro $P \subset R^n$. Si omitimos la restricción de integridad de un PLE resulta un PL que llamaremos relajación lineal del PLE.

El PLE es un problema NP – duro, pues una gran cantidad de problemas NP – duros conocidos se pueden formular como un PLE. Los métodos más utilizados para resolver el PLE son los llamados métodos de "Branch and Bound", y el modo de hacer más eficientes estos algoritmos consiste en encontrar cotas superiores e inferiores al valor óptimo de la función objetivo lo más ajustadas posible. Otros métodos más recientes, como el de Branch and Cut y Branch and Price, han demostrando su gran potencial al resolver problemas específicos de PLE.

Las siguientes referencias son, entre otras, excelentes textos de Programación Lineal y Programación Lineal Entera: Dantzig (1963), Garfinkel y Nemhauser (1972), Bazaraa y Jarvis (1977), Chvátal (1983), Schrijver (1986) y Nemhauser y Wolsey (1988).

1.6. Combinatoria Poliédrica

En esta sección resumiremos la idea fundamental de la aproximación poliédrica a la resolución de los Problemas de Optimización Combinatoria, especialmente de los *NP – duros*.

Dado un conjunto E finito (llamado conjunto base) con una función de coste y una familia \mathcal{F} (finita o infinita numerable) de subconjuntos (o listas) de E (llamadas soluciones posibles), el Problema de Optimización Combinatoria, (abreviadamente P. O. C.), con función objetivo lineal, consiste en encontrar un $F^* \in \mathcal{F}$ tal que $c(F^*) = \sum_{e \in F^*} c_e x_e$ sea mínimo (o máximo), donde x_e denota el número de veces que $e \in E$ aparece en F^* .

La aproximación poliédrica a los P. O. C. del tipo anterior comienza con la creación de un poliedro $P_{\mathcal{F}}$, cuyos vértices, posiblemente junto a otros puntos de $P_{\mathcal{F}}$, siguen una correspondencia uno-a-uno con las soluciones posibles de \mathcal{F} . Se define el vector de incidencia x^F de una solución posible F en \mathcal{F} como $x^F = (x_e^F)_{e \in E} \in \mathbb{Z}^{|E|}$, donde x_e^F denota el número de veces que e aparece en F , y entonces definimos un conjunto convexo $P_{\mathcal{F}}$ como: $P_{\mathcal{F}} = \text{convex}\{x^F : F \in \mathcal{F}\}$.

Aunque el conjunto $P_{\mathcal{F}}$ no es un poliedro en general, para la mayor parte de los P.O.C. se puede demostrar que sí lo es. Supondremos pues que $P_{\mathcal{F}}$ es un poliedro. Claramente, cada solución posible F en \mathcal{F} corresponde a un punto de $P_{\mathcal{F}}$ y cada vértice de $P_{\mathcal{F}}$ corresponde a una solución posible en \mathcal{F} . Por lo tanto, para cualquier función de costes c para la que el P.O.C. tiene un valor óptimo finito, podemos encontrar una solución óptima del P.O.C. por medio de un vértice óptimo del *PL*:

$$\min_{x \in P_{\mathcal{F}}} cx \tag{1.1}$$

Lo que hemos hecho ha sido transformar, de un modo natural, el P.O.C. en el *PL* (1.1). Sin embargo, sin tener conocimiento alguno del sistema lineal que describe $P_{\mathcal{F}}$, esta transformación es inútil desde el punto de vista algorítmico. Para abordar el problema (1.1) con las técnicas de la Programación Lineal, es necesario conocer todo (o una parte suficientemente grande de) el sistema lineal que describe $P_{\mathcal{F}}$.

Aunque conociéramos una descripción lineal completa del poliedro $P_{\mathcal{F}}$, no podríamos generar y almacenar todos los datos del *PL* asociado, ya que, aún siendo el sistema lineal finito, cabe esperar que sea extremadamente grande (en muchos problemas, el número de filas crece exponencialmente con el número de variables del problema). Una alternativa a la enumeración de todas las filas conocidas del sistema lineal es resolver periódicamente el siguiente problema:

Problema de Identificación de Facetas (Problema de Separación)

Dado un punto $\bar{x} \in R^{|E|}$ y un poliedro $P_{\mathcal{F}}$, encontrar una desigualdad lineal $fx \leq f_0$ que defina una faceta de $P_{\mathcal{F}}$ que sea violada (es decir, $f\bar{x} > f_0$) por \bar{x} , o demostrar que tal desigualdad no existe ($\bar{x} \in P_{\mathcal{F}}$).

Utilizando el Problema de Identificación de Facetas como subrutina en un código de Programación Lineal, podríamos generar desigualdades (que definen facetas de $P_{\mathcal{F}}$) violadas a medida que las necesitamos para “cortar” los óptimos fraccionarios de los problemas lineales.

El procedimiento iterativo para la resolución de los P.O.C., introducido por Grötschel y Padberg (1985), es:

Método de Relajación para Problemas de Optimización Combinatoria:

1. **Inicialización.** Sea (PL_0) una relajación lineal del P.O.C. que consideramos. Hacer $k = 0$ e ir a (2).
2. **Resolución del PL.** Resolver (PL_k) . Sea x^k una solución óptima de (PL_k) . Ir a (3).
3. **Identificación de facetas.** Resolver el problema de identificación de facetas para x^k y $P_{\mathcal{F}}$.
 - 3.1 Si se encuentran una o más desigualdades (que definan facetas de $P_{\mathcal{F}}$) violadas por x^k , definir (PL_{k+1}) como (PL_k) junto a las desigualdades encontradas. Hacer $k = k + 1$ e ir a (2).
 - 3.2 Si no existe una desigualdad violada, parar.

Si en el paso (2) utilizamos el método simplex, el procedimiento anterior será finito y convergente, y la terminación en (3.2) ocurrirá en una solución óptima del P.O.C, siempre y cuando conozcamos la descripción lineal completa de $P_{\mathcal{F}}$.

Desgraciadamente, la descripción completa del sistema lineal asociado al poliedro $P_{\mathcal{F}}$ se conoce únicamente para unos pocos Problemas de Optimización Combinatoria (por ejemplo, para el problema de encontrar un acoplamiento de peso máximo en un grafo). Además, éste no es el caso de ningún problema *NP-duro*. De hecho, resultados de la Teoría de la Complejidad Algorítmica conducen a pensar que es altamente improbable llegar a un conocimiento completo del sistema lineal que represente el poliedro $P_{\mathcal{F}}$ asociado a un problema NP-duro, como demuestra Papadimitrou (1977).

Sin embargo, los descubrimientos en Combinatoria Poliédrica de la última década han demostrado que incluso un conocimiento parcial del sistema lineal que describe $P_{\mathcal{F}}$ es, muy frecuentemente, fuente de nuevos resultados significativos tanto en el aspecto teórico como en el computacional. Se puede utilizar la descripción del poliedro $P_{\mathcal{F}}$ para mejorar la relajación lineal simple, añadiendo una lista parcial de desigualdades (que definen facetas) en el Método de Relajación. Si durante la aplicación de este método ocurre que no encontramos una desigualdad violada, puede ser porque:

- i) no existe.
- ii) no conocemos la descripción completa de $P_{\mathcal{F}}$.
- iii) no sabemos como generar algorítmicamente una desigualdad conocida.

Entonces, cuando se presente (3.2) en el Método de Relajación, podemos acudir a un esquema del tipo de Branch and Bound o de Branch and Cut, pero con mayor probabilidad de alcanzar el óptimo, ya que tenemos una relajación mucho más ajustada.

Así pues, uno de los retos fundamentales en Combinatoria Poliédrica es encontrar un subsistema suficientemente grande del sistema lineal completo que describe el poliedro $P_{\mathcal{F}}$ para diseñar estrategias eficientes de resolución del problema.

Algunos ejemplos en los que esta aproximación ha llevado a resultados importantes son el Problema del Acoplamiento (Grötschel y Holland 1985), el Problema del Viajante (Grötschel y Padberg 1985 y Padberg y Rinaldi 1990), el Problema de la Ordenación Lineal (Grötschel, Jünger y Reinelt 1984) y el Problema de Rutas de Vehículos (Augerat et al. 1995). Esta aproximación a los P.O.C. es, de hecho, el centro de atención de la Combinatoria Poliédrica. Excelentes artículos sobre el particular son los de Pulleyblank (1983), Grötschel (1984), Hoffman y Padberg (1985) y el libro de Nemhauser y Wolsey (1988).

Capítulo 2

Algunos Problemas de Rutas

En este capítulo, veremos algunos de los problemas de rutas más conocidos, y nos centraremos especialmente en aquéllos más relacionados con el problema del que se ocupa esta tesis, el WGRP. Para cada uno de estos problemas, daremos una formulación como problema de optimización en un grafo, describiremos su complejidad algorítmica y repasaremos las estrategias de solución más importantes aparecidas en la literatura científica, haciendo especial hincapié en el estudio de los poliedros de soluciones asociados.

2.1. Problemas de Rutas por Vértices o Arcos

Básicamente, podemos dividir los problemas de rutas en dos tipos: los Problemas de Rutas por Vértices y los Problemas de Rutas por Arcos.

En los Problemas de Rutas por Vértices, tenemos un conjunto de vértices en un grafo que han de ser visitados por nuestra solución. Estos problemas representarían, por ejemplo, la situación real en la que un viajante debe visitar una serie de ciudades de manera que la distancia recorrida sea mínima. En este caso, se dice que se trata de un problema con demanda en los vértices. El problema más conocido de este tipo es el *Problema del Viajante* (TSP, Traveling Salesman Problem).

Los Problemas de Rutas por Arcos son aquéllos en los que la solución debe recorrer una serie de aristas y/o arcos del grafo, es decir, la demanda está en los enlaces. Podríamos equipararlos al problema que tendría un cartero que quiere recorrer determinadas calles de la ciudad para repartir el correo en ellas, todo



ello recorriendo la menor distancia posible. A este tipo de problemas pertenece el Problema del Cartero Rural (RPP, Rural Postman Problem).

Existe también una serie de problemas más generales en los que se considera la posibilidad de tener que recorrer un conjunto de aristas y/o arcos, así como visitar un conjunto de vértices del grafo. Entre ellos se encuentra, por ejemplo, el *Problema General de Rutas* (GRP, General Routing Problem)

También se pueden clasificar los Problemas de Rutas según el grafo sea **no dirigido** (que representan una situación en la que, por ejemplo, se pueden recorrer las calles o carreteras en ambos sentidos con el mismo coste), **dirigido** (para representar una situación, por ejemplo, en la que calle sólo puede ser recorrida en un sentido) o **mixto** (para representar el caso en el que algunas calles pueden ser recorridas en un sólo sentido y otras en ambos). Otra variante de estos problemas lo constituyen los que llamamos Problemas de Rutas **con Viento** (o *Windy*, concepto introducido por Minieka (1979) para el Windy Postman Problem, WPP), en los que el grafo es no dirigido pero cada arista (i, j) tiene asociados dos costes c_{ij}, c_{ji} , no necesariamente iguales, que representan el coste de atravesar la arista en cada sentido. Notar que los problemas de tipo *Windy* generalizan los problemas sobre un grafo no dirigido, dirigido o mixto.

Frecuentemente, la zona de demanda es tan grande que no puede ser atendida por un solo vehículo. En estos casos, el problema que se plantea es diseñar una ruta para cada vehículo, de modo que se satisfaga la demanda global. De este modo, también se pueden clasificar los problemas de rutas según sean una o varias las rutas a diseñar.

Por último, cualquiera de estos Problemas de Rutas puede estar complementado con algunas condiciones adicionales, como por ejemplo las *ventanas de tiempo*, dentro de las cuales el servicio debe ser realizado, *relaciones de precedencia* de algunos servicios frente a otros, *giros* o transiciones de un enlace a otro *prohibidos*, etc. . . .

A modo de resumen, la tabla 2.1 representa una clasificación de los Problemas de Rutas clásicos con un único vehículo, donde cada problema, especialmente los problemas con demandas en los enlaces, puede ser considerado sobre un grafo no dirigido, dirigido, mixto o con viento.

El Problema del Viajante, **TSP**, es sin duda el problema de rutas de vehículos que más ha sido estudiado en la literatura científica.

El planteamiento clásico de este problema consiste en construir un grafo completo y exigir que la ruta pase exactamente una vez por cada uno de los vértices del grafo. Karp (1972) demostró que el TSP es un problema NP-duro.

El Problema del Viajante Gráfico (**GTSP**, Graphical Traveling Salesman

| | Demandas en los vértices | Demandas en los enlaces |
|------------|-----------------------------------|--------------------------------|
| 1 vehículo | TSP Traveling Salesman Problem | CPP Chinese Postman Problem |
| | GTSP Graphical TSP | RPP Rural Postman Problem |
| | GRP General Routing Problem | |

Tabla 2.1: Clasificación de los Problemas de Rutas.

Problem), es una variante del TSP introducida por Fleischmann (1985, 1988) y Cornuejols, Fonlupt y Naddef (1985) en la cual el grafo no tiene por qué ser completo y la ruta debe pasar por cada vértice al menos una vez.

Las principales ventajas de esta formulación frente a la formulación clásica del TSP son que el poliedro de soluciones es de dimensión completa y que, en general, son necesarias menos variables, pues se trabaja directamente sobre el grafo que modeliza la red viaria, que normalmente dista mucho de ser un grafo completo.

En cuanto a los *Problemas de Rutas por Arcos*, tenemos el Problema del Cartero Chino (CPP, Chinese Postman Problem), planteado por el matemático chino Meigu Guan, donde la cuestión es determinar una ruta cerrada de coste mínimo que recorra cada enlace del grafo al menos una vez. Edmonds (1965a) demostró que en un grafo no dirigido este problema es resoluble polinómicamente. También lo es en el caso dirigido y se les puede considerar, por lo tanto, problemas resueltos. En cambio, el CPP sobre un grafo mixto y el Windy CPP son problemas NP-duros. El Rural Postman Problem, RPP o Problema del Cartero Rural, es una generalización del problema anterior en el que no hay que pasar por todos los enlaces del grafo, sino solamente por un subconjunto de ellos que llamamos enlaces requeridos. En general, el RPP es NP-duro tanto sobre un grafo no dirigido (Lenstra & Rinnooy-Kan, 1976) como sobre un grafo dirigido o mixto y, por lo tanto, también sobre un grafo con Viento (WRPP, Windy Rural Postman Problem).

Finalmente, el General Routing Problem, GRP, o Problema de Rutas Generalizado, también definido por Orloff (1974), es el caso más general de problemas de rutas sobre un grafo con un único vehículo, pues la demanda se puede encontrar tanto en algunos enlaces como en algunos vértices del grafo. Este problema

es una generalización de los anteriores CPP, RPP y GTSP. Por lo tanto, el GRP es un problema NP-duro.

Hay una gran cantidad de problemas del mundo real que pueden ser formulados como Problemas de Rutas. Por ejemplo, la mayor parte de organismos públicos o empresas privadas han de plantearse problemas relacionados con el reparto del correo, recogida de basuras, limpieza, inspección o mantenimiento de calles, carreteras o redes eléctricas, distribución de todo tipo de productos, mercancías o mensajes, visitas a clientes, transporte de personal, etc. Es por esto que un gran número de Problemas de Rutas han sido estudiados en los últimos 30 años, consiguiéndose un significativo avance tanto en la formulación de los problemas como en el diseño, análisis e implementación de algoritmos de resolución. Los trabajos de Bodin, Golden, Assad y Ball (1983), Benavent, Campos, Corberán y Mota (1983) proporcionan una primera panorámica general sobre los problemas de Rutas por Arcos, Bodin y Golden (1981) ofrecen una clasificación más detallada de estos problemas, y Lenstra y Rinnooy-Kan (1981) discuten su complejidad. Más recientes, los trabajos de Eiselt, Gendreau y Laporte (1995aa, 1995bb), Assad y Golden (1995), y el libro editado por Dror (2000) actualizan gran parte del trabajo realizado sobre los Problemas de Rutas por Arcos.

En las siguientes secciones, describiremos los Problemas de Rutas por Arcos con un único vehículo, cuyas definiciones aparecen de un modo natural según la demanda esté en todos los enlaces (arcos o aristas) de un grafo, en algunos de los enlaces, o incluso simultáneamente en algunos vértices y en algunos enlaces.

2.2. El Problema del Cartero Chino

Consideremos el problema en el que un cartero debe repartir correo en todas las calles de la ciudad. Diferenciaremos varios problemas según el grafo sobre el que esté definido sea no dirigido, dirigido o mixto.

2.2.1. El Problema del Cartero Chino en un Grafo No Dirigido (CPP)

En este caso, las aristas del grafo pueden recorrerse en uno u otro sentido con el mismo coste. El CPP (no dirigido) se define del modo siguiente:

Consideremos un grafo no dirigido $G = (V, E)$ con un coste c_e asociado a cada arista $e \in E$. Debemos encontrar un tour en G que recorra cada arista de E al menos una vez, con coste total mínimo.

Es fácil ver que, para que exista una solución óptima finita, hemos de suponer que el grafo G es conexo y que los costes c_e son no negativos. El primer estudio sobre este problema fue hecho por el matemático chino Guan (1962) al diseñar el recorrido que debía realizar un cartero para repartir la correspondencia (de ahí el nombre de cartero chino), aunque sus orígenes se remontan al famoso Problema de los Puentes de Königsberg (Euler, 1953), de 1736. Un tratamiento exhaustivo del CPP desde el punto de vista de la Teoría de Grafos puede encontrarse en Fleischner (1990, 1991).

El CPP está muy relacionado con el problema de encontrar un acoplamiento perfecto de mínimo coste, puesto que a partir de G , añadiendo aristas que unan los vértices impares, podemos obtener un grafo Euleriano (Edmonds y Johnson, 1973; Christofides, 1973). Edmonds y Johnson (1973) resolvieron el CPP utilizando una adaptación del algoritmo *blossom* para el problema de acoplamiento perfecto (Edmonds, 1965b). Demostraron, pues, que el CPP es resoluble polinómicamente y dieron una descripción completa del poliedro asociado.

En el contexto del problema del acoplamiento perfecto, Grötschel y Holland (1985) implementaron un algoritmo de planos de corte que demostró ser tan eficiente como los algoritmos existentes de tipo combinatorio. Así pues, su procedimiento puede ser también considerado como un método exacto para resolver el CPP.

2.2.2. El Problema del Cartero Chino en un Grafo Dirigido (DCPP)

En este caso, los arcos del grafo pueden recorrerse solamente en un sentido. El DCPP se define del modo siguiente:

Consideremos un grafo dirigido $G = (V, A)$ con un coste c_a asociado a cada arco $a \in A$. Encontrar un tour en G que recorra cada arco de A al menos una vez, con coste total mínimo.

Es fácil ver que, para que exista una solución óptima finita, hemos de suponer que el grafo G es fuertemente conexo y que no existe ningún ciclo de coste negativo. Este último supuesto lo sustituiremos por el de que los costes c_a sean no negativos.

Bajo las condiciones anteriores, un grafo Euleriano de mínimo coste puede obtenerse resolviendo un Problema de Transporte: para cada vértice i , sea S_i el número de arcos entrando en i menos el número de arcos que salen de i . Sea S el conjunto de vértices i con $S_i > 0$ y T el conjunto de vértices i con $S_i < 0$. El DCPP puede plantearse como el problema de encontrar el plan de transporte

de coste mínimo entre las plantas S y los destinos T (Liebling, 1970; Edmonds y Johnson, 1973).

Así pues, el DCPD es también resoluble polinómicamente y se conoce una descripción completa de su poliedro asociado.

2.2.3. El Problema del Cartero Chino en un Grafo Mixto (MCPP)

Consideramos ahora el caso en el que las aristas del grafo pueden recorrerse en uno u otro sentido con el mismo coste, mientras que los arcos pueden recorrerse solamente en un sentido. El MCPP puede definirse del modo siguiente:

Sea $G = (V, E, A)$ un grafo mixto con un coste c_e asociado a cada enlace $e \in E \cup A$. Encontrar un tour en G que recorra cada enlace de $E \cup A$ al menos una vez con coste total mínimo.

No es difícil ver que, para que exista una solución óptima finita, hemos de suponer que el grafo G es fuertemente conexo y que no existe ningún ciclo de coste negativo. Como en el caso dirigido, este último supuesto lo sustituiremos por el de que los costes c_e sean no negativos.

Para este problema NP-duro (Papadimitriou, 1976), Christofides, Benavent, Campos, Corberán y Mota (1984) propusieron una formulación con una variable por cada arco, dos variables por cada arista (representando el número de veces que es atravesada en cada dirección) y una variable por cada vértice. Desarrollaron un algoritmo de Branch and Bound en el que dos cotas inferiores diferentes, obtenidas de relajar lagrangianamente dos tipos de restricciones, eran calculadas en cada nodo del árbol de ramificación. Usando este algoritmo resolvieron un conjunto de 34 instancias generadas aleatoriamente con $7 \leq |V| \leq 50$, $3 \leq |A| \leq 85$ y $4 \leq |E| \leq 39$.

El MCPP es un caso especial del Windy Postman Problem (WPP), del que hablaremos en la próxima sección. Como veremos allí, Grötschel y Win (1992) propusieron un procedimiento para resolver el WPP que puede ser también aplicado al MCPP. Las 9 instancias del MCPP que intentaron resolver lo fueron óptimamente. Los tamaños son $52 \leq |V| \leq 172$, $31 \leq |A| \leq 116$ y $37 \leq |E| \leq 154$.

Nobert y Picard (1996) han propuesto una formulación para el MCPP en la que hay una única variable asociada a cada arista de E . La solución entera, pues, no especifica la dirección en que las aristas son recorridas. Se imponen restricciones que aseguran que el grafo solución será par y equilibrado. Su principal contribución es la demostración de que el Problema de Separación para las restricciones de equilibrio puede ser resuelto polinómicamente. Basándose en

ello, diseñan un algoritmo de planos de corte con el que resuelven 148 de las 180 instancias probadas. Los tamaños de estas instancias están en los rangos $10 \leq |V| \leq 169$, $2 \leq |A| \leq 2876$ y $15 \leq |E| \leq 1849$.

Es importante notar que, si G es par, el MCPP es resoluble en tiempo polinómico (Edmonds y Johnson, 1973). El algoritmo exacto para el caso en que G sea par es la base de dos heurísticos para el caso general sugeridos por Edmonds y Johnson (1973) y desarrollados y mejorados por Frederickson (1979). El primer algoritmo equivaldría a transformar primero G en un grafo par y aplicar luego el método antes citado. El segundo algoritmo puede considerarse como la versión inversa: primero obtiene un grafo simétrico y luego lo mantiene simétrico mientras lo hace par. Frederickson (1979) demostró que ambos algoritmos tienen radio del peor caso 2, mientras que si aplicamos ambos algoritmos y seleccionamos la mejor solución, el radio del peor caso es, ahora, $5/3$. Recientemente, Raghavachari y Veerasamy (1998) han propuesto una modificación al algoritmo conjunto anterior cuyo radio del peor caso es $3/2$. También recientemente, Corberán, Martí y Sanchis (2002) han desarrollado un algoritmo tipo GRASP que ha proporcionado soluciones posibles alejadas, en media, un 1.12% de cotas inferiores en 225 instancias con $50 \leq |V| \leq 200$ y $120 \leq |E \cup A| \leq 600$.

El poliedro del MCPP como caso particular del MRPP y del MGRP, ha sido estudiado en Romero (1997), Corberán, Romero y Sanchis (2003) y Corberán, Mejía y Sanchis (2005aa).

2.2.4. El Problema del Cartero Chino Con Viento (WPP)

Supondremos ahora que las aristas del grafo pueden recorrerse en uno u otro sentido pero con costes diferentes según el sentido del recorrido. El WPP puede definirse como:

Consideremos un grafo no dirigido $G = (V, E)$ con dos costes asociados a cada arista $(i, j) \in E$: c_{ij} , que es el coste de atravesar dicha arista desde i hasta j y c_{ji} , que es el coste de atravesar la arista de j a i . Debemos hallar un tour en G que recorra cada arista de E al menos una vez con coste total mínimo.

Una condición suficiente para que exista una solución óptima finita es que el grafo G sea conexo y que todos los costes sean no negativos.

Este problema, propuesto por Guan (1984), comprende como casos particulares a los tres anteriores y, por tanto, es NP-duro.

Un buen estudio sobre el WPP puede encontrarse en la tesis de Win (1989). Allí se demuestra que el WPP se puede resolver en tiempo polinómico en algunos supuestos, entre ellos si G es Euleriano. Grötschel y Win (1992) proponen

un algoritmo de planos de corte para el WPP basado fundamentalmente en la identificación de restricciones de corte impar violadas por la solución lineal actual. Este Problema de Separación es resoluble en tiempo polinómico (Padberg y Rao, 1982), lo que les permite resolver instancias del WPP de entre 52 y 264 vértices y de entre 78 y 489 aristas. La solución lineal así obtenida proporciona la solución óptima del WPP en 31 de las 36 instancias probadas.

2.3. El Problema del Cartero Rural

Supongamos ahora que el cartero anterior debe repartir correo sólo en algunas calles de la ciudad, y no en todas. De nuevo planteamos este problema según el grafo sea no dirigido, dirigido o mixto.

2.3.1. El Problema del Cartero Rural en un Grafo No Dirigido (RPP)

El Problema del Cartero Rural No Dirigido se define del modo siguiente:

Consideremos un grafo no dirigido $G = (V, E)$ con un coste c_e asociado a cada arista $e \in E$ y sea $E_R \subseteq E$ un subconjunto no vacío de aristas de G , a las que llamaremos *requeridas*. El objetivo es encontrar un tour en G que recorra cada arista de E_R al menos una vez con coste total mínimo.

Este problema fue definido por primera vez por Orloff (1974). Lenstra y Rinnooy-Kan (1976) probaron que el RPP es un problema NP-duro. Orloff señaló que la complejidad del RPP parece aumentar con el número de componentes conexas del subgrafo inducido por las aristas requeridas. Este hecho coincide con la observación de Frederickson (1979) de que existe un algoritmo recursivo exacto para el RPP que es exponencial en el número de tales componentes.

Frederickson (1979) y Christofides, Campos, Corberán y Mota (1981) propusieron un algoritmo aproximado de resolución para el RPP basado en el cálculo de un árbol generador de mínimo coste que conecta las componentes conexas inducidas por las aristas requeridas y en la resolución, posteriormente, de un problema de acoplamiento mínimo sobre los vértices impares (respecto de las aristas requeridas y las aristas del árbol calculado anteriormente). Este algoritmo heurístico, similar al propuesto por Christofides (1976) para el TSP, tiene asegurada una desviación máxima respecto del valor óptimo del 50 % (Benavent, Campos, Corberán y Mota, 1985).

Recientemente, Hertz, Laporte, y Nanchen (1999) han desarrollado varios

procedimientos de mejora de soluciones posibles generadas con algoritmos heurísticos constructivos. Estos procedimientos, que pueden extenderse fácilmente a grafos dirigidos y mixtos, producen buenos resultados computacionales.

Christofides, Campos, Corberán y Mota (1981) también describieron un algoritmo de Branch and Bound para la resolución exacta del RPP. Las cotas inferiores son calculadas a partir del problema del árbol generador de mínimo coste que surge como una relajación lagrangiana. La ramificación en el árbol de búsqueda se realiza sobre aristas que unen componentes conexas inducidas por las aristas requeridas. Con este algoritmo resuelven problemas de hasta 84 vértices, 180 aristas y 74 aristas requeridas.

Sanchis (1990), Corberán y Sanchis (1994, 1998) y Letchford (1997, 1999) estudiaron el poliedro del RPP y su generalización, el Problema General de Rutas (GRP), que será introducido más adelante. Algunos de estos resultados serán comentados a lo largo de la presente memoria.

Basándose en la descripción antes citada, Corberán, Letchford y Sanchis (2001) diseñaron un algoritmo de planos de corte capaz de resolver las 26 instancias de Christofides et al. (1981) y Corberán y Sanchis (1994), así como las 92 instancias del RPP generadas por Hertz, Laporte y Nanchen (1999).

Otra aproximación fue propuesta por Ghiani y Laporte (2000). Caracterizando un subconjunto de soluciones óptimas, estos autores formularon el RPP usando únicamente variables 0/1. De esta forma implementaron un algoritmo de Branch and Cut con el que consiguieron muy buenos resultados en instancias con hasta 350 vértices. Esta aproximación ha sido continuada recientemente con éxito por Reinelt y Theis (2004).

Basándose en otra formulación diferente, Fernández, Meza, Garfinkel y Ortega (2003) propusieron un algoritmo de Branch and Cut con el que han demostrado que esta nueva aproximación es competitiva con las propuestas anteriormente.

2.3.2. El Problema del Cartero Rural en un Grafo Dirigido (DRPP)

El Problema del Cartero Rural Dirigido se define de la siguiente manera:

Consideremos un grafo dirigido $G = (V, A)$ con un coste c_a asociado a cada arco $a \in A$ y sea $A_R \subseteq A$ un subconjunto no vacío de arcos de G que llamaremos *requeridos*. Buscamos un tour en G que recorra cada arco de A_R al menos una vez con coste total mínimo.

Como en el caso no dirigido, si el subgrafo inducido por A_R no es conexo, el DRPP es NP-duro. Basándose en la formulación del problema dada por Christofides, Campos, Corberán y Mota (1986), Savall (1990) define el poliedro de soluciones del DRPP y estudia algunas de sus propiedades. También, de forma independiente, ha sido estudiado por Gun (1993).

Campos y Savall (1995) realizan un estudio comparativo de tres heurísticos existentes para el DRPP: uno propuesto por ellos, otro por Ball y Magazine (1988) y un tercero por Christofides, Campos, Corberán y Mota (1986). La idea general de los tres heurísticos es similar. Básicamente, o cambia el orden de los pasos efectuados o la forma de conectar las componentes conexas inducidas por los arcos requeridos. El estudio del peor caso en el algoritmo heurístico de Christofides et al. (1986) es presentado en Benavent, Campos, Corberán y Mota (1983).

Respecto de la resolución exacta del DRPP, Christofides et al. (1986) describen un algoritmo de Branch and Bound cuya cota superior se obtiene con el heurístico antes mencionado; la cota inferior es obtenida relajando lagrangianamente las condiciones de simetría y calculando una arborescencia de mínimo coste.

Algunos de los resultados en Romero (1997) y Corberán, Romero y Sanchis (2003) para el RPP en un grafo mixto (que veremos después) se pueden aplicar directamente al DRPP. Por lo tanto, el algoritmo de planos de corte diseñado para el MRPP podría ser utilizado también para resolver el DRPP.

2.3.3. El Problema del Cartero Rural en un Grafo Mixto (MRPP)

El Problema del Cartero Rural Mixto se define de la siguiente forma:

Consideremos un grafo mixto $G = (V, E, A)$ con un coste c_e asociado a cada enlace $e \in E \cup A$ y sea $E_R \subseteq E$ un subconjunto no vacío de aristas de G que llamaremos requeridas y $A_R \subseteq A$ un subconjunto no vacío de arcos de G que llamaremos requeridos. Hemos de hallar un tour en G que recorra cada enlace de $E_R \cup A_R$ al menos una vez con coste total mínimo.

Una condición suficiente para que el MRPP tenga solución óptima finita es que el grafo sea fuertemente conexo y no existan aristas con coste negativo ni ciclos con coste total negativo. Nosotros supondremos que todos los costes asociados a enlaces del grafo son no negativos.

Nótese que, si $A_R = \phi$ ($E_R = \phi$), el MRPP se reduce al RPP (DRPP) y que si $E_R = E$ y $A_R = A$, obtenemos el MCPP (que a su vez generaliza al CPP y al

DCPP). Es evidente, pues, que el MRPP es, en general, un problema NP-duro.

Algunos trabajos recientes sobre el MRPP son los de Romero (1997) y Corberán, Romero y Sanchis (2003). En ellos, dada una instancia del MRPP, ésta es transformada en una nueva en la que cada vértice $v \in V$ es incidente con, al menos, un enlace requerido y donde cada arista $e = (i, j)$ no requerida es reemplazada por dos arcos no requeridos (i, j) y (j, i) y, así, $E = E_R$.

En su artículo sobre el poliedro del MGRP, del cual el poliedro del MRPP es un caso particular, Corberán, Romero y Sanchis (2003) presentan algunos resultados computacionales con un algoritmo de planos de corte sobre 100 instancias generadas aleatoriamente con $20 \leq |V| \leq 100$, $15 \leq |E| \leq 200$, $55 \leq |A| \leq 350$ y hasta 15 componentes R-conexas. Este algoritmo produce la solución óptima en 28 de ellas y, en promedio, la cota obtenida dista menos de 0.5% del coste de una solución posible obtenida por los procedimientos heurísticos descritos en Corberán, Martí y Romero (2000).

En un trabajo posterior, Corberán, Mejía y Sanchis (2005aa) amplían el estudio del poliedro del MRPP y mejoran el algoritmo de planos de corte anterior.

2.4. El Problema del Cartero Rural con Viento (WRPP)

El Problema del Cartero Rural con Viento se puede definir como:

Dado un grafo no dirigido $G = (V, E)$, con dos costes c_{ij}, c_{ji} asociados a cada arista $(i, j) \in E$ representando el coste de recorrer dicha arista en cada sentido, sea $E_R \subseteq E$ un subconjunto de aristas requeridas. El objetivo es hallar un tour en G que recorra al menos una vez todas las aristas de E_R (sin importar en qué sentido) con coste total mínimo.

Una condición suficiente para que el WRPP tenga solución óptima finita es que el grafo sea conexo y no existan ciclos de coste total negativo. De nuevo, supondremos que los costes de las aristas son todos no negativos.

En el caso particular en que $E_R = E$, tenemos el WPP y, de la misma forma en la que el WPP generaliza al CPP, DCP y MCPP, el RPP, DRPP y MRPP son casos particulares del WRPP. Así pues, el WRPP es un problema NP-duro que generaliza a la mayor parte de los Problemas de Rutas por Arcos con un único vehículo.

Los únicos trabajos realizados sobre el WRPP son los de Benavent, Ca-



rrotta, Corberán, Sanchis y Vigo (2003) y Benavent, Corberán, Piñana, Plana y Sanchis (2005). En el primero de ellos, se propone una formulación entera del problema, así como una cota inferior (obtenida mediante un algoritmo de planos de corte) y varios algoritmos heurísticos constructivos. Los resultados computacionales aportados demuestran que tanto las cotas inferiores como las superiores son buenas en instancias de tamaño medio. El segundo trabajo propone distintos algoritmos heurísticos para la resolución aproximada del WRPP. Entre ellos, dos heurísticos constructivos que son modificaciones de los algoritmos propuestos por Benavent et al. (2003). La aleatorización de algunos de los algoritmos constructivos iniciales y de sus modificaciones dan lugar a algoritmos metaheurísticos de tipo Multiarranque (Multi-Start) que, como se demuestra, proporcionan muy buenas soluciones posibles en tiempos razonables de computación. Finalmente, en Benavent et al. (2004) se propone un sofisticado algoritmo de Búsqueda Dispersa (Scatter Search) que proporciona excelentes soluciones, aunque con tiempos mayores de computación.

Puesto que el WRPP es un caso particular del Windy General Routing Problem (WGRP), objeto de esta tesis, todos los resultados que se expondrán en los capítulos siguientes son de aplicación directa al WRPP.

2.5. El Problema General de Rutas

Es el caso más general de entre los Problemas de Rutas con un único vehículo: aquél en el que la demanda de servicio se puede encontrar tanto en los enlaces como en los vértices del grafo.

2.5.1. El Problema General de Rutas sobre un Grafo No Dirigido (GRP)

El GRP no dirigido se define del modo siguiente:

Consideremos un grafo no dirigido $G = (V, E)$ con un coste c_e asociado a cada arista $e \in E$. Consideremos un subconjunto de aristas que llamaremos requeridas $E_R \subseteq E$ y un subconjunto de vértices que llamaremos requeridos $V_R \subseteq V$. Debemos encontrar un tour en G que recorra cada arista requerida y pase por cada vértice requerido al menos una vez, con coste total mínimo.

Este problema fue introducido por Orloff (1974) y es NP-duro (Lenstra y Rinnooy-Kan, 1976). Incluye como casos especiales la mayoría de los problemas sobre grafos no dirigidos presentados anteriormente. En concreto, si $E_R = E$

tenemos el CPP, si $V_R = \phi$ resulta el RRP y si $E_R = \phi$ y $V_R = V$ se reduce al Problema del Viajante Gráfico (GTSP).

La formulación como PLE del GRP es exactamente la misma que para el RPP. Así pues, todas las desigualdades válidas para el RPP lo son para el GRP, y viceversa. El problema y su poliedro han sido extensamente estudiados en Corberán y Sanchis (1994, 1998) y Letchford (1997, 1999). Corberán, Letchford y Sanchis (2001) han desarrollado un algoritmo de planos de corte que ha producido excelentes resultados computacionales en instancias del RPP, GTSP y GRP "puras". Nótese también que los algoritmos propuestos por otros autores para la resolución del RPP (comentados en aquella sección) pueden ser también aplicables a la resolución del GRP.

2.5.2. El Problema General de Rutas en un Grafo Mixto (MGRP)

Este problema se define de la siguiente forma:

Sea G un grafo mixto $G = (V, E, A)$ con costes no negativos asociados a sus enlaces. Consideremos un subconjunto de arcos y un subconjunto de aristas, $A_R \subseteq A$ y $E_R \subseteq E$ respectivamente, que llamaremos requeridos y un subconjunto $V_R \subseteq V$ de vértices requeridos. Buscamos un tour en G que pase por cada enlace y vértice requerido al menos una vez y con coste total mínimo.

Evidentemente, este problema NP-duro contiene a todos los problemas presentados previamente (excepto al WPP) como caso particular. Al igual que sucede con su versión no dirigida, la formulación como PLE del MGRP y MRPP coincide, por lo que, como ya hemos expuesto en el apartado dedicado al MRPP, los resultados obtenidos para uno de los problemas son aplicables al otro.

La descripción del poliedro del MGRP hecha en los trabajos de Corberán, Romero y Sanchis (2003) y Corberán, Mejía y Sanchis (2005aa) es bastante completa, como demuestran los resultados computacionales obtenidos con un algoritmo de planos de corte descrito en el segundo de los artículos mencionados. El algoritmo es capaz de resolver óptimamente 65 de las 81 instancias generadas (con hasta 214 componentes conexas generadas por los vértices y enlaces requeridos) y alcanzar una desviación media respecto del óptimo (en aquellas instancias no resueltas pero con óptimo conocido) de un 0.16 %.

Algunos de los resultados descritos en esos artículos serán generalizados aquí al Problema General de Rutas con Viento (WGRP), objeto de esta tesis.

Capítulo 3

El Problema General de Rutas con Viento

En este capítulo, presentaremos con detalle el Problema General de Rutas con Viento (WGRP de aquí en adelante). Expondremos sus aplicaciones, tanto teóricas como prácticas, y propondremos una formulación lineal entera, así como distintas caracterizaciones de sus soluciones. También introduciremos el estudio de su poliedro de soluciones, comenzando por su dimensión. Veremos que prácticamente todas las desigualdades conocidas tienen una serie de características en común que dan lugar al concepto de *configuración*, *grafo de configuración* y *desigualdad de configuración*. Por último, presentaremos un *teorema de lifting*, que ayudará posteriormente a demostrar que todas estas desigualdades de configuración inducen facetas del poliedro de soluciones del WGRP.

3.1. Descripción del problema

El Problema General de Rutas con Viento se define sobre un grafo no dirigido y tiene las siguientes características principales:

- Cada arista tiene dos costes asociados (que pueden ser distintos), que representan el coste de recorrerla en cada uno de los dos sentidos posibles.
- Existe un subconjunto de aristas, que llamaremos *aristas requeridas*, que hemos de recorrer al menos una vez, sin importar el sentido.

- Existe un subconjunto de vértices, que llamaremos *vértices requeridos*, que hemos de visitar al menos una vez.

El objetivo es hallar la forma de recorrer dicho grafo con el menor coste posible, atravesando todas las aristas requeridas al menos una vez y visitando todos los vértices requeridos. Éste es un problema de gran interés, tanto por sus aplicaciones prácticas en la vida real, como por su utilidad teórica, ya que contiene como casos particulares a la mayoría de problemas de rutas con un único vehículo.

3.1.1. Definición

Sea $G = (V, E)$ un grafo no dirigido, $E_R \subseteq E$ un subconjunto de aristas requeridas y $V_R \subseteq V$ un subconjunto de vértices requeridos. Para cada arista $(i, j) \in E$, sea c_{ij} el coste de recorrer dicha arista del vértice i al vértice j , y c_{ji} el coste de recorrerla en el sentido contrario. El WGRP consiste en hallar un tour de coste mínimo que pase por todas las aristas requeridas y todos los vértices requeridos al menos una vez.

3.1.2. Casos particulares

Como hemos comentado anteriormente, el WGRP contiene a muchos problemas de rutas como casos particulares:

- **WRPP:** Si el conjunto de vértices requeridos es vacío, $V_R = \emptyset$, tenemos el Problema del Cartero Rural con Viento.
- **GRP:** Si $c_{ij} = c_{ji} \forall (i, j) \in E$, tenemos el Problema de General de Rutas en un grafo no dirigido, que, a su vez, contiene como casos particulares los siguientes problemas:
 - **RPP:** Si el conjunto de vértices requeridos es vacío, $V_R = \emptyset$ (Problema del Cartero Rural).
 - **CPP:** Si el conjunto de vértices requeridos es vacío, $V_R = \emptyset$, y todas las aristas son requeridas, $E_R = E$ (Problema del Cartero Chino).
- **DGRP:** El GRP sobre un grafo dirigido puede plantearse como un WGRP donde por cada arco (i, j) con coste C tenemos una arista (i, j) con costes $c_{ij} = C$ y $c_{ji} = \infty$. A su vez, el DGRP contiene al DRPP y al DCPP como casos particulares.

- **MGRP**: El GRP sobre un grafo mixto puede plantearse como un WGRP, donde las aristas están representadas por aristas con ambos costes iguales, y los arcos, al igual que en el DGRP, por aristas con coste infinito en un sentido. El MGRP contiene también como casos particulares al MRPP y al MCPP.
- **WPP**: Si todas las aristas son requeridas, $E = E_R$, tenemos el Problema del Cartero Chino con Viento.
- **GATSP**: Si el conjunto de aristas requeridas es vacío, $E_R = \emptyset$, y todos los vértices son requeridos, $V_R = V$, tenemos el Problema del Viajante Gráfico Asimétrico.
- **GTSP**: Si el conjunto de aristas requeridas es vacío, $E_R = \emptyset$, todos los vértices son requeridos, $V_R = V$, y $c_{ij} = c_{ji} \forall (i, j) \in E$, tenemos el Problema del Viajante Gráfico.

3.1.3. Aplicaciones

El estudio del WGRP surge ante la necesidad de resolver un problema real concreto, cuya resolución, como veremos a continuación, se puede plantear como uno de los problemas incluidos como casos particulares en el WGRP, el Problema del Cartero Rural con Viento (WRPP).

En determinadas estructuras arquitectónicas complejas, como por ejemplo puentes (figura 3.1), en las que una gran cantidad de vigas se entrecruzan para formar un gran armazón, resulta imprescindible una revisión periódica de la superficie de todas estas vigas con objeto de comprobar si han sufrido algún tipo de deterioro que pueda poner en peligro la estructura. Dada la complejidad de estas estructuras, y puesto que a menudo se encuentran a gran altura sobre el nivel del suelo, resulta extremadamente peligroso, si no imposible, que este trabajo sea llevado a cabo directamente por personas. Para realizar esta tarea se han diseñado (y se sigue trabajando en nuevos) robots teledirigidos (figura 3.2, ver Balaguer, Giménez, Pastor, Padrón y Abderrahim, 2000), capaces de moverse por la superficie de las vigas que componen la estructura. Estos robots están dotados de una cámara que transmite las imágenes a un monitor, mediante el cual una persona puede comprobar si la superficie de la viga presenta alguna grieta o anomalía que pueda resultar peligrosa y haya que reparar. Sin embargo, estos robots no tienen una autonomía ilimitada, sino que el tiempo que pueden estar en movimiento viene determinado por la batería que llevan incorporada. Resulta, por tanto, importante diseñar el recorrido del robot de manera que éste realice la inspección de la estructura con el menor consumo energético posible.

Este problema puede plantearse como un problema de Rutas por Arcos de la siguiente manera:



Figura 3.1: Aplicaciones. Estructuras a inspeccionar

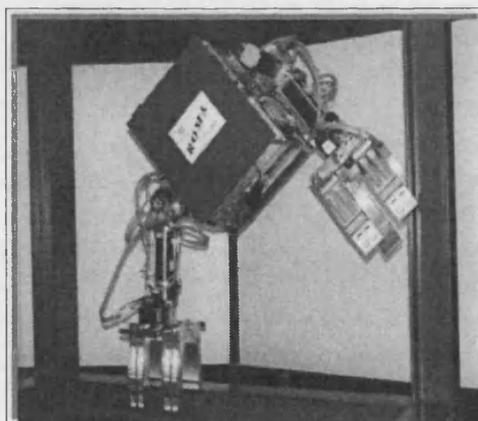


Figura 3.2: Aplicaciones. Robot teledirigido

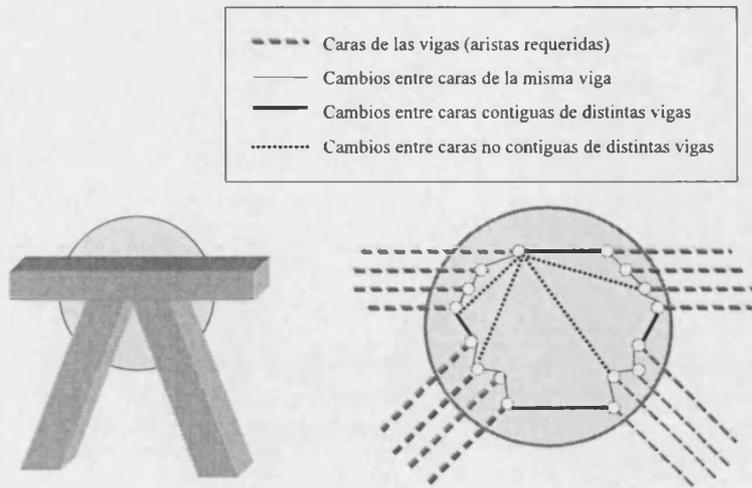


Figura 3.3: Representación del problema como WRPP

Construiremos un grafo que represente la estructura a inspeccionar y los posibles movimientos del robot. Cada viga tiene cuatro caras que hemos de inspeccionar, cada una de las cuales vendrá representada por una arista en el grafo. Estas aristas comienzan y acaban cada una en un nodo distinto.

Cuando el robot llega al punto de unión entre varias vigas, puede realizar alguno de los siguientes movimientos: seguir recorriendo la misma viga pero cambiar de cara, continuar avanzando por la siguiente viga por la cara inmediatamente adyacente a la que estaba recorriendo o cambiar a una cara no adyacente de otra viga. Cada uno de estos movimientos supone un gasto diferente de energía, y vendrá representado por una arista uniendo los nodos que representan los extremos de las caras entre las que se desplaza el robot. En la figura 3.3 podemos ver un ejemplo de cómo se genera el grafo.

Cada arista tiene asociados dos costes, que representan el consumo de energía al recorrer la cara o realizar el correspondiente movimiento. Dichos costes están asociados a los dos sentidos posibles en que se puede recorrer la viga o llevar a cabo el movimiento, y pueden ser distintos, ya que, por ejemplo, el consumo de energía del robot no es el mismo al recorrer una viga vertical hacia arriba o hacia abajo.

Las aristas que representan las caras de las vigas han de ser recorridas todas necesariamente, puesto que no podemos dejar ninguna cara sin ser inspeccionada. Estas aristas serán, por tanto, las aristas requeridas. Por otra parte, tenemos otra serie de aristas, las que representan los movimientos del robot en los puntos de unión, que pueden ser recorridas o no, según convenga. Éstas serán las aristas no requeridas.

El objetivo es encontrar un tour de coste mínimo, es decir, una secuencia de aristas, que nos indique el recorrido que debe realizar el robot de manera que el gasto de energía sea mínimo, pero sin dejar ninguna cara por inspeccionar.

De esta manera hemos modelizado el problema como un problema de rutas por arcos, en concreto un WRPP, ya que se cumplen las siguientes condiciones:

- Hemos de recorrer una serie de aristas obligatoriamente (*Postman*).
- Existe una serie de aristas no requeridas (*Rural*).
- Cada arista tiene dos costes asociados, uno para cada sentido, que pueden ser distintos (*Windy*).

3.1.4. Soluciones del problema

Una solución del WGRP puede caracterizarse de las siguientes formas:

- Un tour, es decir, un camino cerrado (*closed walk*). Se trata de una sucesión de aristas que nos da el orden y la dirección en que éstas son recorridas. Una misma arista puede aparecer más de una vez en un tour, es decir, podemos recorrerla varias veces, si es necesario. El tour debe contener todas las aristas y vértices requeridos.
- Un grafo dirigido $G^* = (V^*, A^*)$. Dicho grafo contendrá una copia del arco (i, j) por cada vez que la solución recorra la arista (i, j) de i a j . Para que el grafo sea una solución válida del problema ha de ser fuertemente conexo, simétrico (el número de arcos que entran en cada nodo ha de ser igual al número de arcos que salen), por cada arista requerida (i, j) ha de tener al menos un arco entre i y j (sin importar la dirección) y ha de contener todos los vértices requeridos, $V_R \subseteq V^*$. Aunque esta representación no indica el orden en que se han de recorrer las aristas, éste puede ser hallado en tiempo polinómico.
- Un vector $x \in \mathbb{R}^{2|E|}$ en el que a cada arista (i, j) le corresponden dos componentes x_{ij} y x_{ji} del vector, cuyo valor representa el número de veces que dicha arista es recorrida por el tour de i a j y de j a i respectivamente, o el número de veces que el arco (i, j) o (j, i) aparece en el grafo G^* anterior. Esta representación tampoco indica el orden en que se recorren las aristas.

En lo que sigue, nos referiremos también como tour a la tercera de estas representaciones y será la que utilizaremos más comúnmente.

3.2. Formulación

En primer lugar, queremos hacer notar que si un tour recorre una determinada arista, también visita sus dos vértices terminales. Así, los vértices incidentes con aristas requeridas serán necesariamente visitados por cualquier tour. En lo que sigue y sin pérdida de generalidad los consideraremos como vértices requeridos.

3.2.1. Simplificación

Aunque una instancia del WGRP puede tener vértices no requeridos, es decir, puede darse $V_R \neq V$, cualquier instancia puede ser simplificada hasta transformarla en una equivalente donde todos los vértices sean requeridos, $V_R = V$. Para ello, dado un vértice u no requerido (que, por la generalización anterior, no es incidente con ninguna arista requerida), por cada par posible de aristas $(u, i), (u, j)$ incidentes con u añadimos una nueva arista no requerida (i, j) con costes $c_{ij} = c_{iu} + c_{uj}$ y $c_{ji} = c_{ju} + c_{ui}$. Una vez hecho esto para todo par posible de aristas incidentes con u , podemos eliminar el nodo u y todas las aristas incidentes con él. Si, al añadir las nuevas aristas, obtenemos aristas paralelas (es decir, hemos añadido una arista entre un par de nodos que ya estaban conectados), sustituiremos todas estas aristas paralelas por una sola cuyos costes asociados sean los menores de los costes de dichas aristas.

De esta manera, obtenemos un grafo $G = (V_R, E)$ donde todos los vértices son requeridos. De ahora en adelante, trabajaremos siempre con grafos de este tipo, aunque, por simplicidad, seguiremos denotándolos como $G = (V, E)$.

3.2.2. Notación

Si eliminamos de un grafo $G = (V, E)$ las aristas no requeridas, el grafo resultante $G_R = (V, E_R)$ puede no ser conexo. Si el número de componentes conexas de G_R es p , denotaremos por V_1, V_2, \dots, V_p y llamaremos *componentes R-conexas* o *R-sets* a los subconjuntos de vértices en V asociados a cada componente conexa de G_R . Nótese que un vértice requerido no incidente con ninguna arista requerida forma una componente conexa por sí solo, y que $V = \bigcup_{i=1}^p V_i$ y $V_i \cap V_j = \emptyset$ si $i \neq j$. Recordemos que si S y T son conjuntos de vértices, $(S : T)$ representa el conjunto de aristas con un vértice en S y el otro en T . Del mismo modo, $(S : T)_R$ y $(S : T)_{NR}$ representarán, respectivamente, las aristas requeridas y no requeridas de $(S : T)$. También denotaremos $\delta(S)$ ($\delta_R(S)$, $\delta_{NR}(S)$) al conjunto de aristas (requeridas, no requeridas) con un extremo en S y otro en $V \setminus S$. Dado un conjunto de aristas L y un vector $x \in \mathbb{R}^{2|E|}$, denotaremos $x(L)$

a la suma de las componentes de x asociadas a las aristas de L , es decir:

$$x(L) := \sum_{(i,j) \in L} (x_{ij} + x_{ji})$$

3.2.3. Formulación Lineal Entera

A continuación, presentamos una formulación lineal entera para el WGRP:

$$\text{Min} \sum_{(i,j) \in E} c_{ij}x_{ij} + c_{ji}x_{ji} \quad (3.1)$$

$$\text{s.a. :} \quad x_{ij} + x_{ji} \geq 1 \quad \forall (i,j) \in E_R \quad (3.2)$$

$$\sum_{(i,j) \in \delta(i)} (x_{ij} - x_{ji}) = 0 \quad \forall i \in V \quad (3.3)$$

$$\sum_{(i,j): i \in S, j \in V \setminus S} x_{ij} \geq 1 \quad \forall S = \bigcup_{k \in Q} V_k, \quad Q \subset \{1, \dots, p\} \quad (3.4)$$

$$x_{ij}, x_{ji} \geq 0 \quad \text{enteras} \quad \forall (i,j) \in E \quad (3.5)$$

La función objetivo (3.1) es, obviamente, la suma de los costes de las aristas recorridas. Las ecuaciones (3.2) son las que obligan a que cada arista requerida sea recorrida al menos una vez, no importa en qué sentido, y reciben el nombre de *desigualdades de obligatoriedad*. Las desigualdades (3.3) reciben el nombre de ecuaciones de *conservación de flujo* o de *simetría*. Obligan a que el número de veces que el tour llega a un nodo sea igual al número de veces que sale de él. Las desigualdades de *conectividad* (3.4) evitan la formación de subtours (camino dirigido cerrado desconectado del resto de arcos de la solución). Con esta desigualdad, se obliga a que, si tenemos una partición del conjunto de nodos en dos subconjuntos no vacíos, de manera que no haya ninguna arista requerida entre ellos, la solución pase al menos una vez de un subconjunto a otro. Por último, (3.5) obliga a que las variables sean no negativas y enteras. Estas desigualdades reciben el nombre de *desigualdades triviales*.

Obsérvese que, aunque explícitamente no aparezca la condición de que todos los vértices requeridos han de ser visitados, ésta no es necesaria. Los vértices requeridos incidentes con alguna arista requerida serán obligatoriamente visitados gracias a la condición (3.2), mientras que aquéllos que no sean incidentes con ninguna arista requerida formarán por sí solos un R -set, con lo que las restricciones de conectividad (3.4) garantizan que cualquier solución del problema también los visitará.

3.3. El poliedro de soluciones

Definimos el poliedro de soluciones del WGRP, al que denotaremos por $WGRP(G)$, como la envoltura convexa del conjunto de soluciones posibles del problema, es decir:

$$WGRP(G) := \text{convex}\{x \in \mathbb{R}^{2|E|} : x \text{ es un tour para el WGRP en } G\}$$

3.3.1. Dimensión del poliedro

A continuación, demostraremos que el poliedro de soluciones del WGRP tiene dimensión $2|E| - |V| + 1$.

Lema 3.3.1 *La envoltura afín de $WGRP(G)$ es el conjunto de vectores de $\mathbb{R}^{2|E|}$ que cumplen las ecuaciones (3.3):*

$$\text{afín}(WGRP(G)) = \{x \in \mathbb{R}^{2|E|} : x \text{ cumple (3.3)}\}$$

Dem: Consideremos el sistema formado por las ecuaciones (3.3). Veamos que, en caso de existir otra ecuación satisfecha por todos los puntos de $WGRP(G)$, ésta es una combinación lineal de las ecuaciones de dicho sistema.

Sea $a^T x = b$ una ecuación satisfecha por todos los puntos de $WGRP(G)$. Primero, vamos a ver algunas propiedades del vector a .

Consideremos un tour x y una arista cualquiera (i, j) del conjunto E . Entonces, el vector $y \in \mathbb{R}^{2|E|}$ definido como $y_{ij} = x_{ij} + 1$, $y_{ji} = x_{ji} + 1$, $y_{kl} = x_{kl}$ en otro caso, es también un tour en G , puesto que no hemos hecho más que añadir un ciclo a un tour. Así pues, se cumplirá $a^T x = a^T y = b$. Por lo tanto, $0 = a^T y - a^T x = a_{ij} + a_{ji}$.

Sea z un ciclo dirigido de longitud al menos 3. Si añadimos este ciclo al tour x , obtenemos otro tour $x+z$ que cumplirá $a^T x = a^T(x+z) = b$, de lo que se deduce que $a^T z = 0$.

Vamos a definir un vector $\lambda \in \mathbb{R}^{|V|}$ del modo siguiente. Sea T un árbol generador de G , e i_0 un nodo cualquiera de V . Damos a λ_{i_0} un valor arbitrario. Supongamos que ya hemos dado un valor a λ_i . Sea (i, j) una arista de T incidente con el nodo i , tal que λ_j todavía no tiene ningún valor asignado. Entonces, definimos $\lambda_j = \lambda_i - a_{ij}$.

De esta manera, una vez elegido el primer valor λ_{i_0} arbitrariamente, se asigna un valor a cada uno de los demás elementos de λ .

Veamos que $a_{ij} = \lambda_i - \lambda_j \quad \forall (i, j) \in E$. Si $(i, j) \in T$, y el valor de λ_i había sido asignado antes que el de λ_j , entonces, por la propia definición que hemos hecho de λ_j , la afirmación se cumple. Si λ_j había sido elegido antes, sabemos que $a_{ji} = \lambda_j - \lambda_i$, y, como hemos visto antes que $a_{ij} + a_{ji} = 0$, se deduce que la igualdad también es cierta. En el caso en que $(i, j) \notin T$, podemos

encontrar un camino dirigido P de j a i formado por aristas del árbol T . Si a este camino le añadimos la arista (i, j) , tenemos un ciclo C . Como $a^T C = 0$, y $a_{kl} = \lambda_k - \lambda_l \quad \forall (k, l) \in P$, es fácil deducir que $a_{ij} = \lambda_i - \lambda_j$.

Así pues, tenemos que $a_{ij} = \lambda_i - \lambda_j \quad \forall (i, j) \in E$. Si M es la matriz de coeficientes del sistema de ecuaciones (3.3), tenemos que $a^T = \lambda^T M$, ya que cada columna de M es un vector con todos sus elementos nulos excepto dos, en los que tiene un 1 y un -1.

De esta manera, queda probado que la envoltura afin del poliedro es el conjunto de vectores de $\mathbb{R}^{2|E|}$ que cumplen las ecuaciones (3.3). \blacklozenge

Teorema 3.3.2 *La dimensión de $WGRP(G)$ es $2|E| - |V| + 1$.*

Dem: Es fácil ver que la matriz de coeficientes M de las ecuaciones (3.3) tiene rango $|V| - 1$, por lo que $\dim(WGRP(G)) \leq 2|E| - (|V| - 1)$. Por otro lado, el lema anterior implica que $\dim(WGRP(G)) \geq 2|E| - |V| + 1$, con lo que tenemos $\dim(WGRP(G)) = 2|E| - |V| + 1$. \blacklozenge

3.4. Propiedades generales del poliedro

Antes de estudiar familias concretas de desigualdades que inducen faceta del poliedro $WGRP(G)$, vamos a estudiar algunas propiedades comunes a una gran parte de ellas.

En primer lugar, veremos que la mayoría de estas desigualdades pueden describirse mediante una determinada estructura definida sobre G y que llamaremos *configuración* (Naddef y Rinaldi, 1991), que lleva asociada un grafo ‘reducido’ que resulta de comprimir el grafo original G y que llamaremos *grafo de configuración*. Después veremos condiciones (teorema de ‘lifting’) para que, si una desigualdad induce faceta del poliedro del WGRP asociado al grafo de configuración entonces también induzca faceta de $WGRP(G)$.

3.4.1. Desigualdades de configuración

Consideremos, por ejemplo, las desigualdades de conectividad

$$\sum_{(i,j): i \in S, j \in V \setminus S} x_{ij} \geq 1 \quad \forall S = \bigcup_{k \in Q} V_k, \quad Q \subset \{1, \dots, p\}$$

que, como veremos más adelante, inducen faceta de $WGRP(G)$. Estas desigualdades están descritas por una partición del conjunto de vértices V en dos conjuntos, S y $V \setminus S$, de modo que todas las variables asociadas a aristas en $E(S)$

y en $E(V \setminus S)$ tienen coeficiente cero en la desigualdad, todas las variables x_{ij} con $i \in S, j \in V \setminus S$ tienen el mismo coeficiente (igual a 1) y todas las variables correspondientes a la dirección opuesta, x_{ji} , también tienen el mismo coeficiente (0 en este caso).

De este modo, una desigualdad de conectividad está descrita por un grafo con dos únicos vértices, $\{1, 2\}$, correspondientes a S y $V \setminus S$, y una única arista $(1, 2)$ no requerida con dos coeficientes $c(1, 2) = 1, c(2, 1) = 0$. Este es el que llamaremos *grafo de configuración* de la desigualdad, G_C . El teorema de ‘lifting’ que veremos más adelante nos da condiciones para asegurar que, si la desigualdad de conectividad en $G_C, x_{12} \geq 1$, induce faceta de $\text{WGRP}(G_C)$, lo que en este caso es muy fácil de demostrar, entonces la desigualdad de conectividad original en G induce faceta de $\text{WGRP}(G)$.

Veremos que hay un gran número de desigualdades que inducen faceta de $\text{WGRP}(G)$ que pueden ser expresadas como desigualdades de configuración. Definimos a continuación este concepto formalmente.

Una *configuración* \mathcal{C} en un grafo $G = (V, E)$ se define como un par (\mathcal{B}, c) , donde $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_r\}$ es una partición de V tal que cada subgrafo $G(\mathcal{B}_i)$ es conexo y c es una función definida sobre $\mathcal{B} \times \mathcal{B}$ tal que no existe ningún ciclo cerrado $\mathcal{B}_i, \mathcal{B}_j, \dots, \mathcal{B}_m, \mathcal{B}_i$ con c -coste total negativo.

Asociado a una configuración \mathcal{C} tenemos un *grafo de configuración* $G_C = (V^{\mathcal{C}}, E^{\mathcal{C}})$ con las siguientes características:

- Por cada subconjunto $\mathcal{B}_i \in \mathcal{B}$, existe un vértice en $V^{\mathcal{C}}$, al que llamaremos B_i .
- Por cada arista requerida $(u, v) \in E_R$ entre dos subconjuntos $\mathcal{B}_i, \mathcal{B}_j \in \mathcal{B}$ (i.e., $u \in \mathcal{B}_i, v \in \mathcal{B}_j$), tenemos una arista requerida (B_i, B_j) entre B_i y B_j .
- Por cada par de subconjuntos $\mathcal{B}_i, \mathcal{B}_j \in \mathcal{B}$ tales que $(\mathcal{B}_i : \mathcal{B}_j)_{NR} \neq \emptyset$ tenemos una única arista no requerida (B_i, B_j) .

Este grafo de configuración puede interpretarse como el resultado de comprimir en G cada subconjunto \mathcal{B}_i en un solo vértice B_i , y las aristas no requeridas paralelas, es decir, aquéllas que conecten el mismo par de subconjuntos, en una única arista no requerida, manteniendo todas las aristas requeridas entre subconjuntos, que podrán ser aristas en paralelo en el grafo de configuración (ver figura 3.4). Para simplificar la notación, y siempre que no de lugar a confusión, denotaremos por B_i indistintamente al subconjunto de nodos $\mathcal{B}_i \subset V$ en el grafo original G y al vértice $B_i \in V^{\mathcal{C}}$ del grafo de configuración.

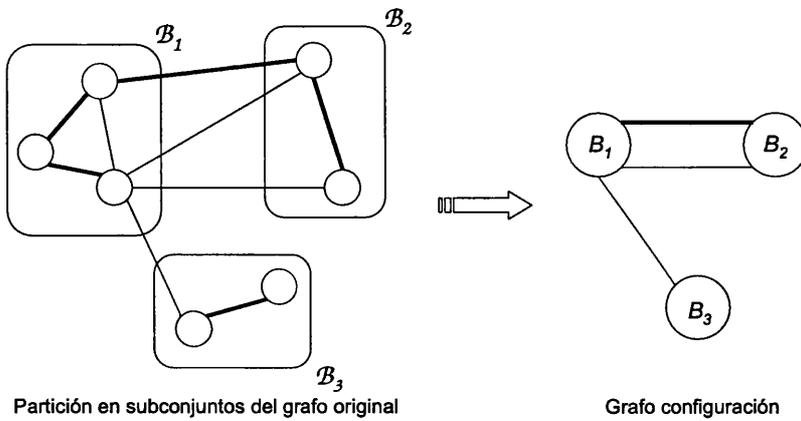


Figura 3.4: Ejemplo de construcción de un grafo de configuración

Una configuración \mathcal{C} define una desigualdad que llamaremos *desigualdad de configuración*,

$$\sum_{(i,j) \in E} (c(i,j)x_{ij} + c(j,i)x_{ji}) \geq c_0$$

donde:

- $c(i,j) = c(B_i, B_j)$ y $c(j,i) = c(B_j, B_i) \quad \forall (i,j) \in (B_i : B_j)$.
- $c(i,j) = c(j,i) = 0 \quad \forall (i,j) \in E(B_i), i = 1, \dots, r$
- c_0 es el coste del tour de c -coste mínimo del WGRP definido en el grafo $G_{\mathcal{C}}$

Para ver que muchas de las desigualdades que inducen faceta de $\text{WGRP}(G)$ son de configuración introducimos el siguiente concepto: **Def.** Sea x un tour para el WGRP en G . Diremos que x utiliza la variable x_{ij} de modo extra si, al quitar una copia del arco (i,j) en el grafo dirigido asociado a x y reemplazarla por un camino de i a j , el tour obtenido sigue siendo un tour para el WGRP.

Notar que este concepto coincide con el concepto de ‘recorrer una arista sin servirla’. Si $e = (i,j)$ es una arista no requerida (por tanto no hay que servirla), decir que x utiliza la variable x_{ij} de modo extra es sencillamente decir que $x_{ij} \geq 1$. Sin embargo, si $e = (i,j)$ es una arista requerida, hay que servirla exactamente una de las veces en que es recorrida, y puede recorrerse de i a j o de j a i . Así, x utiliza la variable x_{ij} de modo extra si $x_{ij} \geq 1$ y $x_{ij} + x_{ji} \geq 2$.

Lema 3.4.1 Sea $F(x) \geq c_0$ una desigualdad no trivial ni de obligatoriedad que induce una faceta de $WGRP(G)$.

(a) Para cualquier x_{ij} asociada a una arista $e = (i, j)$ (requerida o no requerida), existe un tour x que cumple $F(x) = c_0$ y $x_{ij} \geq 1$.

(b) Para cualquier arista requerida $e = (i, j) \in E_R$, existe un tour x que cumple $F(x) = c_0$ y $x_{ij} + x_{ji} \geq 2$.

Dem: Si no se cumpliera (a), todo tour x cumpliendo $F(x) = c_0$ también cumpliría $x_{ij} = 0$. Es decir,

$$\{x \text{ WGRP-tour en } G : F(x) = c_0\} \subseteq \{x \text{ WGRP-tour en } G : x_{ij} = 0\}$$

y, por lo tanto, o bien $F(x) \geq c_0$ es equivalente a una desigualdad trivial, o no induce una faceta de $WGRP(G)$.

Si no se cumpliera (b), todo tour x cumpliendo $F(x) = c_0$ también cumple $x_{ij} + x_{ji} = 1$. Es decir,

$$\{x \text{ WGRP-tour en } G : F(x) = c_0\} \subseteq \{x \text{ WGRP-tour en } G : x_{ij} + x_{ji} = 1\}$$

y, por lo tanto, o bien $F(x) \geq c_0$ es equivalente a una desigualdad de obligatoriedad, o no induce una faceta de $WGRP(G)$. ♦

Este lema implica que para cualquier x_{ij} asociado a una arista no requerida $e = (i, j)$, existe un tour para el WGRP en G que satisface $F(x) = c_0$ y que utiliza la variable x_{ij} de modo extra. También implica que, para cualquier x_{ij} asociado a una arista requerida $e = (i, j)$, existe un tour para el WGRP en G que satisface $F(x) = c_0$ y que utiliza la variable x_{ij} (aunque no necesariamente de modo extra). Finalmente, para cualquier arista requerida $e = (i, j)$, $x_{ij} + x_{ji} \geq 2$ implica que existe un tour para el WGRP en G que satisface $F(x) = c_0$ y que utiliza, al menos, una de las variables x_{ij} o x_{ji} de modo extra. Sin embargo puede ocurrir que, para alguna variable x_{ij} asociada a una arista requerida $e = (i, j) \in E_R$, no exista ningún tour cumpliendo $F(x) = c_0$ que la utilice de modo extra.

Teorema 3.4.2 Sea $F(x) \geq c_0$ una desigualdad que induce faceta de $WGRP(G)$. Si para cualquier x_{ij} asociada a una arista $e \in E_R$, existe un tour x que cumple $F(x) = c_0$, $x_{ij} \geq 1$ y $x_{ij} + x_{ji} \geq 2$ (es decir, utiliza x_{ij} de modo extra), entonces $F(x) \geq c_0$ es una desigualdad de configuración.

Dem: Sea $E^0 = \{e = (i, j) \in E : c(i, j) = c(j, i) = 0\}$ el conjunto de todas las aristas del grafo cuyas dos variables asociadas no aparecen en la desigualdad. Consideremos el subgrafo de G inducido por estas aristas, $G^0 = (V, E^0)$. Veamos que las componentes conexas de este grafo definen la partición del conjunto V que forma la configuración. Sea S el conjunto de nodos de una componente conexa de G^0 .

Si existiese una arista $e = (i, j) \in E(S)$ con $c(i, j) < 0$, añadiéndole un camino en $E^0(S)$ desde j hasta i obtendríamos un ciclo de c -coste negativo. Entonces, añadiendo a un tour cualquiera para el WGRP en G un número suficiente

de copias de este ciclo obtendríamos un tour para el WGRP que violaría la desigualdad, lo cual es absurdo.

Si existiese una arista no requerida $e = (i, j) \in E(S)$ con $c(i, j) > 0$, sea x^0 un tour para el WGRP en G que utilice x_{ij} de modo extra y que satisfaga $F(x^0) = c_0$. Reemplazando en el grafo dirigido asociado a x^0 el arco (i, j) por un camino en $E^0(S)$ desde i hasta j obtenemos un tour x^* para el WGRP en G con $F(x^*) < F(x^0) = c_0$, lo que es una contradicción.

Supongamos que existe una arista $e = (i, j) \in \gamma(S)$ con $c(i, j) > 0$. Por la hipótesis del teorema, existe un tour x^0 tal que $cx^0 = c_0$ y $x_{ij}^0 + x_{ji}^0 \geq 2$ y $x_{ij} \geq 1$ (es decir, utiliza x_{ij} de modo extra), y, procediendo como antes, llegamos a una contradicción.

Así pues, si $\mathcal{B} = \{B_1, B_2, \dots, B_r\}$ son los conjuntos de nodos de las componentes conexas de G^0 , los subgrafos $G(B_k)$ son conexos y $c(i, j) = c(j, i) = 0$, $\forall e = (i, j) \in E(B_k)$, $k = 1, 2, \dots, r$.

Supongamos ahora que existen dos aristas $e = (i, j)$, $f = (u, v)$ con $i, u \in B_p$ y $j, v \in B_q$, $p \neq q$, tales que $c(i, j) > c(u, v)$. Sea x^0 un tour para el WGRP que utilice x_{ij} de modo extra y que cumpla $F(x^0) = c_0$ (si e es no requerida, dicho tour existe siempre; si e es requerida, dicho tour existe por la hipótesis del teorema). Reemplacemos en el grafo dirigido asociado a x^0 esta copia de e por un camino de i a u en $G(B_p)$, la arista f recorrida de u a v más un camino de v a j en $G(B_q)$ (los caminos existen pues $G(B_p)$ y $G(B_q)$ son conexos) y obtendremos un tour x^* para el WGRP en G con $F(x^*) < F(x^0) = c_0$, lo que es una contradicción. Por tanto, todas las variables asociadas a aristas en $E(B_p : B_q)$ en la dirección de B_p a B_q tienen el mismo coeficiente en la desigualdad.

Finalmente, si existiese un ciclo cerrado $B_i, B_j, B_k, \dots, B_m, B_i$ con c -coste total negativo, añadiendo a cualquier tour un número suficiente de copias de tal ciclo obtendríamos otro tour que violaría la desigualdad. \blacklozenge

Una configuración \mathcal{C} sobre G puede también considerarse una configuración sobre el grafo comprimido G_C y, por lo tanto, también define una desigualdad sobre este grafo:

$$\sum_{(i,j) \in E^{\mathcal{C}}} (c(B_i, B_j)x_{ij} + c(B_j, B_i)x_{ji}) \geq c_0$$

donde x_{ij} representa la variable asociada a la arista (B_i, B_j) recorrida de B_i a B_j . Además, cualquier tour x para el WGRP en G puede ser comprimido para convertirlo en un tour x^C para el WGRP en G_C y ambos tendrán el mismo c -coste. Por otra parte, un tour x^c para el WGRP en G_C puede convertirse fácilmente en un tour x válido para el WGRP en G con el mismo c -coste, puesto que los subgrafos $G(B_i)$ son conexos. Por lo tanto, si la desigualdad de configuración es válida para WGRP(G), también lo es para WGRP(G_C).

Mediante el siguiente teorema demostraremos que toda desigualdad de configuración que define una faceta de WGRP(G_C), define también una faceta de WGRP(G).

Teorema 3.4.3 *Sea C una configuración en G . Si la desigualdad de configuración asociada a C como configuración en el grafo G_C define una faceta de $WGRP(G_C)$, entonces la desigualdad de configuración asociada a C como configuración en el grafo G define una faceta de $WGRP(G)$.*

Dem: Basta demostrar el resultado para el grafo, que por simplicidad llamaremos G , resultante de reemplazar en G_C el supernodo B_1 (por ejemplo) por un grafo conexo $G(B_1)$. Por comodidad, denotemos por $F(x) \geq c_0$ indistintamente a la desigualdad de configuración en los grafos G_C y G .

Sea $K = \dim(WGRP(G_C))$ y supongamos que $c_0 \neq 0$. Como $F(x) \geq c_0$ define una faceta de $WGRP(G_C)$, existen K tours en G_C linealmente independientes que cumplen $F(x) = c_0$. Si $c_0 = 0$, tendríamos $K-1$ vectores, pero la prueba sería análoga.

Al expandir el supernodo B_1 al grafo $G(B_1)$, aparecen Q aristas nuevas no requeridas entre nodos de $G(B_1)$ y supernodos de G_C distintos de B_1 . Llamemos G'_C al grafo G_C más Q aristas $\hat{e}_1, \hat{e}_2, \dots, \hat{e}_Q$ no requeridas, incidentes con B_1 y paralelas a alguna arista no requerida de G_C .

Para cada arista \hat{e}_i , $i = 1, 2, \dots, Q$, vamos a construir dos tours en G'_C . Sea $e_j = (B_1, B_q)$ la arista original de G_C a la que es paralela \hat{e}_i . Por el lema anterior existe un tour en G_C que cumple $F(x) = c_0$ y que recorre la arista desde B_1 hasta B_q y otro que la recorre en sentido contrario. Si cambiamos en esos dos tours la arista e_j por la arista \hat{e}_i , obtenemos dos nuevos tours en G'_C que cumplen $F(x) = c_0$. Así, si consideramos los K tours anteriores en G_C como tours en G'_C , tenemos $K+2Q$ tours en G'_C que cumplen $F(x) = c_0$ y que son linealmente independientes. Estos tours pueden ser completados con aristas del grafo $G(B_1)$ y obtenemos $K+2Q$ tours en G que también cumplen $F(x) = c_0$, pues las variables asociadas a las aristas en $G(B_1)$ no aparecen en la desigualdad. Sean estos tours $x^1, x^2, \dots, x^{K+2Q}$.

Consideremos ahora el WGRP definido en el subgrafo $G(B_1)$. Si $M = \dim(WGRP(G(B_1)))$, existen y^1, y^2, \dots, y^M tours para el WGRP en $G(B_1)$ linealmente independientes. Entonces, $x^1+y^1, x^1+y^2, \dots, x^1+y^M$ son tours para el WGRP en G que cumplen también $F(x) = c_0$ (de nuevo porque las variables asociadas a aristas en $G(B_1)$ no aparecen en la desigualdad). Así, tenemos $K+2Q+M$ tours para el WGRP en G que cumplen $F(x) = c_0$. Para ver que son linealmente independientes, expresamos estos tours como filas y restamos x^1 a las M últimas, resultando la matriz siguiente, que es de rango completo:

$$\begin{pmatrix} G'_C & G(B_1) \\ \begin{array}{c|c} x^1 & \\ \vdots & * \\ x^{K+2Q} & \end{array} \\ \hline \begin{array}{c|c} 0 & y^1 \\ \vdots & \vdots \\ & y^M \end{array} \end{pmatrix}$$

Por tanto, los tours $x^1, \dots, x^{K+2Q}, x^1 + y * 1, \dots, x^1 + y^M$ son linealmente independientes. Falta ver que el número $K + 2Q + M$ de tours linealmente independientes coincide con la dimensión de $\text{WGRP}(G)$. Notemos que $K = 2|E^c| - |V^c| + 1$, $M = 2|E_{G(B_1)}| - |V_{G(B_1)}| + 1$ y Q es el número de aristas no requeridas en paralelo en G'_c . Sumando, obtenemos

$$K + 2Q + M = 2|E| - (|V_{G_c}| - 1 + |V_{G(B_1)}|) + 1 = 2|E| - |V| + 1 = \dim(\text{WGRP}(G)). \quad \blacklozenge$$

3.4.2. Desigualdades de configuración débil

Las desigualdades triviales y las de obligatoriedad no son de configuración. Muchas de las demás desigualdades que inducen faceta de $\text{WGRP}(G)$ sí lo son, aunque no todas. Veamos a continuación cómo son las desigualdades que no cumplen la hipótesis del teorema 3.4.2, que son las únicas que pueden no ser de configuración.

Sea $F(x) \geq c_0$ una desigualdad que induce faceta de $\text{WGRP}(G)$. Supongamos que para alguna arista requerida $e = (i, j)$ no existe ningún tour x cumpliendo $F(x) = c_0$ que utilice x_{ij} de modo extra, es decir, cumpliendo que $x_{ij} \geq 1$ y que $x_{ij} + x_{ji} \geq 2$. Entonces, todos los tours que cumplen $F(x) = c_0$ cumplen también que, o bien $x_{ij} = 0$, o bien $x_{ij} + x_{ji} = 1$. Es decir, que cada tour que está en la faceta de $\text{WGRP}(G)$ definida por $F(x) \geq c_0$, también está, bien en la faceta definida por la desigualdad trivial $x_{ij} \geq 0$, o bien en la faceta definida por la desigualdad de obligatoriedad $x_{ij} + x_{ji} \geq 1$.

Si todos esos tours están en la faceta trivial, la desigualdad $F(x) \geq c_0$ es equivalente a la desigualdad trivial. Del mismo modo, si todos están en la faceta definida por $x_{ij} + x_{ji} \geq 1$, la desigualdad $F(x) \geq c_0$ es equivalente a la desigualdad de obligatoriedad. En otro caso, existen tours en ambas facetas, es decir, existen al menos dos tours x^1, x^2 cumpliendo $F(x^1) = F(x^2) = c_0$ tales que $x^1_{ij} = 0$, $x^1_{ji} \geq 2$ y $x^2_{ij} = 1$, $x^2_{ji} = 0$ respectivamente.

Supongamos que dicha arista requerida $e = (i, j)$ está entre los supernodos B_1 y B_2 y que entre ellos hay otra arista (u, v) tal que para sus dos variables asociadas x_{uv} , x_{vu} sí existen tours cumpliendo $F(x) = c_0$ que las utilizan de modo extra. Nótese que, por el lema 3.4.1, esto es cierto siempre si (u, v) es no requerida. Sean $c(u, v)$, $c(v, u)$ los coeficientes de estas variables en la desigualdad.

También por el lema 3.4.1, puesto que no existe ningún tour que recorra x_{ij} de modo extra, ha de existir un tour que utilice x_{ji} de modo extra. Sea x^0 ese tour que cumple $F(x^0) = c_0$. Cambiando una copia de (j, i) por un camino de j a v , más una copia de (v, u) , más un camino de u a i , obtenemos otro tour x^1 que cumple $F(x^1) = F(x^0) - c(j, i) + c(v, u) = c_0 - c(j, i) + c(v, u)$. Como

también ha de satisfacer $F(x^1) \geq c_0$, se cumplirá $c(j, i) \leq c(v, u)$. Procediendo de modo análogo con un tour que utilice x_{vu} de modo extra se obtiene que $c(v, u) \leq c(j, i)$, luego $c(j, i) = c(v, u)$. Nótese que si (p, q) y (r, s) son dos aristas entre B_1 y B_2 para las que existen tours cumpliendo $F(x) = c_0$ que utilizan x_{pq} , x_{qp} , x_{rs} y x_{sr} de modo extra, un razonamiento similar nos lleva a la conclusión de que los coeficientes de la desigualdad coinciden para ambos sentidos, es decir, $c(p, q) = c(r, s)$ y $c(q, p) = c(s, r)$.

Si tomamos ahora un tour que utilice x_{uv} de modo extra y cambiamos una copia de (u, v) por un camino de u a i , más una copia de (i, j) , más un camino de j a v , se obtiene un tour x^2 que recorre (i, j) de modo extra, luego debe cumplir que $F(x^2) > c_0$ (pues no existe ningún tour utilizando x_{ij} de modo extra que la cumpla con igualdad). Se obtiene así que $c(u, v) < c(i, j)$.

Sean $c(1, 2)$ y $c(2, 1)$ los coeficientes correspondientes a las variables asociadas a todas las aristas de $(B_1 : B_2)$ para las que sí existen tours cumpliendo $F(x) = c_0$ que las utilizan de modo extra en ambos sentidos. Entonces, si (i, j) es una arista requerida para la que no existe un tour que utilice x_{ij} de modo extra, los coeficientes asociados a las variables x_{ij} , x_{ji} son $c(1, 2) + k$ y $c(2, 1)$, para algún $k > 0$. En consecuencia, los coeficientes asociados a una arista no requerida serán siempre menores o iguales que los de las aristas requeridas, siendo exactamente iguales en al menos uno de los dos sentidos.

Respecto al k anterior, podemos ver que $k \leq c(1, 2) + c(2, 1)$. Sea x^0 un tour cumpliendo $F(x^0) = c_0$ que utiliza la arista x_{ij} (no de modo extra). Si cambiamos (i, j) por dos caminos de i a u , más dos copias de (u, v) , más dos caminos de v a i , más una copia de (j, i) , obtenemos un tour que debe cumplir $F(x) \geq c_0$, luego $c(1, 2) + k \leq c(1, 2) + c(2, 1) + c(1, 2)$.

Una consecuencia importante del análisis anterior de los coeficientes es que pueden existir desigualdades que definen faceta de $\text{WGRP}(G)$ con coeficientes diferentes para aristas entre los mismos subconjuntos B_i de la partición.

Por lo tanto, las únicas desigualdades que inducen faceta de $\text{WGRP}(G)$ que no son de configuración son:

- (1) las triviales,
- (2) las de obligatoriedad
- (3) y desigualdades que están basadas también en una partición $\mathcal{B} = \{B_1, B_2, \dots, B_r\}$ de V con los grafos inducidos $G(B_i)$ conexos pero en donde *algunos* pares de coeficientes asociados a aristas requeridas entre los mismos conjuntos B_i pueden ser distintos.

Estas últimas desigualdades, a las que llamaremos de *configuración débil*,

también tienen asociado un grafo de configuración con las mismas características que las desigualdades de configuración, pues en ellos todas las aristas requeridas, que son las que pueden tener coeficientes distintos, se mantienen.

Más adelante, veremos ejemplos de desigualdades que inducen faceta de $WGRP(G)$ que son de configuración débil.

Como recoge el siguiente teorema, toda desigualdad de configuración débil que define faceta de $WGRP(G_C)$, define también una faceta de $WGRP(G)$.

Teorema 3.4.4 *Sea C una configuración débil en G . Si la desigualdad de configuración débil asociada a C como configuración débil en el grafo G_C define una faceta de $WGRP(G_C)$, entonces la desigualdad de configuración débil asociada a C como configuración débil en el grafo G define una faceta de $WGRP(G)$.*

Dem: La demostración del teorema 3.4.3 puede aplicarse aquí, puesto que los coeficientes de las aristas requeridas no juegan ningún papel en ella. ♦

Capítulo 4

Descripción del poliedro: Facetas naturales

En este capítulo, estudiamos condiciones para que las desigualdades (3.2), (3.4) y (3.5) de la formulación induzcan faceta de $WGRP(G)$. Además, veremos también bajo qué circunstancias las desigualdades R-impares definen facetas del poliedro.

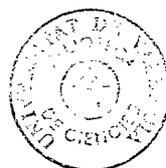
4.1. Desigualdades triviales y de obligatoriedad

En esta sección discutiremos las condiciones bajo las cuales las restricciones triviales (3.5) y de obligatoriedad (3.2) que aparecen en la formulación del WGRP definen facetas de su poliedro asociado.

Teorema 4.1.1 *Sea $e = (i, j) \in E_{NR}$. Si e no es un puente de G , las desigualdades $x_{ij} \geq 0$ y $x_{ji} \geq 0$ definen facetas de $WGRP(G)$.*

Dem: Dado que e no es puente de G , el grafo resultante de eliminar dicha arista, $G \setminus \{e\}$, es conexo. Por lo tanto, $WGRP(G \setminus \{e\}) \neq \emptyset$ y $\dim(WGRP(G \setminus \{e\})) = 2(|E| - 1) - |V| + 1 = K - 2$, donde $K = \dim(WGRP(G))$.

Existen, por tanto, $K - 2$ tours para el WGRP en $G \setminus \{e\}$ linealmente independientes, a los que llamaremos x^1, x^2, \dots, x^{K-2} . Estos tours también serán válidos para el WGRP en G , ya que e es no requerida, y cumplirán $x_{ij} = 0$. Sea C^1 un ciclo dirigido en G que pase exactamente una vez por e en dirección de j a i . Sabemos que dicho ciclo existe puesto que $G \setminus \{e\}$ es conexo, por lo que



existirá un camino de i a j . Entonces, $x^1 + C^1$ es un tour para el WGRP en G , que sigue cumpliendo $x_{ij} = 0$.

Los tours $x^1, \dots, x^{K-2}, x^1 + C^1$ son linealmente independientes, ya que $x^1 + C^1$ tiene una componente, la correspondiente a la arista (i, j) de j a i , que vale uno, cuando en el resto de tours dicha componente vale cero. Así pues, este último tour no puede expresarse como combinación lineal de los anteriores que, además, eran linealmente independientes, con lo que todos lo son.

Tenemos, pues, $K - 1$ tours linealmente independientes cumpliendo $x_{ij} = 0$. Como el término de la derecha de la desigualdad es 0, este número es suficiente para demostrar que la desigualdad $x_{ij} \geq 0$ define una faceta de $\text{WGRP}(G)$.

Procediendo de manera similar, escogiendo en este caso un ciclo C^2 que recorra e de i a j , obtenemos un tour $x^1 + C^2$, que unido a los $K - 2$ tours originales nos proporciona $K - 1$ tours linealmente independientes con $x_{ji} = 0$, por lo que la desigualdad $x_{ji} \geq 0$ también define una faceta de $\text{WGRP}(G)$. \blacklozenge

Teorema 4.1.2 *Sea $e = (i, j) \in E_R$. Si e no es puente de G , las desigualdades $x_{ij} \geq 0$, $x_{ji} \geq 0$ y $x_{ij} + x_{ji} \geq 1$ definen facetas de $\text{WGRP}(G)$.*

Dem: Como e no es puente de G , $G \setminus \{e\}$ es conexo, y existen $K - 2$ tours x^1, x^2, \dots, x^{K-2} linealmente independientes satisfaciendo $x_{ij} = x_{ji} = 0$. Nótese que estos tours en $G \setminus \{e\}$ no lo son en G , ya que no recorren la arista e , que es requerida.

Sea C^1 un ciclo que recorre una vez e de j a i . Entonces, $x^1 + C^1, x^2 + C^1, \dots, x^{K-2} + C^1$ son tours linealmente independientes en G satisfaciendo $x_{ij} = 0$. Consideremos también el tour $x^1 + 2C^1$, que es un tour para el WGRP en G y también cumple $x_{ij} = 0$. Estos $K - 1$ tours son linealmente independientes, puesto que si restamos C^1 de todos ellos se obtienen los vectores $x^1, x^2, \dots, x^{K-2}, x^1 + C^1$, que son linealmente independientes. Por lo tanto, la desigualdad $x_{ij} \geq 0$ induce una faceta de $\text{WGRP}(G)$. Análogamente podemos probar que x_{ji} induce una faceta de $\text{WGRP}(G)$.

Para ver que $x_{ij} + x_{ji} \geq 1$ induce una faceta de $\text{WGRP}(G)$ necesitaremos K tours linealmente independientes cumpliendo $x_{ij} + x_{ji} = 1$, ya que el término de la derecha de la desigualdad no es cero. Sea entonces C^2 un ciclo en G que recorra e exactamente una vez de i a j . Los K tours $x^1 + C^1, x^2 + C^1, \dots, x^{K-2} + C^1, x^1 + C^2, x^1 + x^2 + C^2$ cumplen $x_{ij} + x_{ji} = 1$. Veamos que son linealmente independientes.

Sean $\alpha_1, \alpha_2, \dots, \alpha_K$ coeficientes tales que:

$$\sum_{i=1}^{K-2} \alpha_i (x^i + C^1) + \alpha_{K-1} (x^1 + C^2) + \alpha_K (x^1 + x^2 + C^2) = 0$$

Esto es equivalente a:

$$(\alpha_1 + \alpha_{K-1} + \alpha_K) x^1 + (\alpha_2 + \alpha_K) x^2 + \sum_{i=3}^{K-2} \alpha_i x^i + \sum_{i=1}^{K-2} \alpha_i C^1 + (\alpha_{K-1} + \alpha_K) C^2 = 0 \quad (4.1)$$

La componente correspondiente a la arista e en sentido de i a j vale 1 para C^1 y 0 para todos los vectores x^i y C^2 , por lo que habrá de cumplirse:

$$\sum_{i=1}^{K-2} \alpha_i = 0 \quad (4.2)$$

Del mismo modo, la componente correspondiente a e en sentido de j a i vale 1 en C^2 y 0 en el resto de vectores, por lo que también se cumplirá:

$$\alpha_{K-1} + \alpha_K = 0 \quad (4.3)$$

Así pues, (4.1) queda

$$(\alpha_1 + \alpha_{K-1} + \alpha_K)x^1 + (\alpha_2 + \alpha_K)x^2 + \sum_{i=3}^{K-2} \alpha_i x^i = 0$$

Como el término de la izquierda de la ecuación es una combinación lineal de x^1, x^2, \dots, x^{K-2} , que son linealmente independientes, se cumplirá:

$$\begin{aligned} \alpha_i &= 0 & i = 3, \dots, K-2 \\ \alpha_1 + \alpha_{K-1} + \alpha_K &= 0 \\ \alpha_2 + \alpha_K &= 0 \end{aligned}$$

Esto, unido a (4.2) y (4.3), implica $\alpha_1 = \alpha_2 = \dots = \alpha_K = 0$, con lo que los K tours cumpliendo $x_{ij} + x_{ji} = 1$ son linealmente independientes y la desigualdad de obligatoriedad $x_{ij} + x_{ji} \geq 1$ induce una faceta de $\text{WGRP}(G)$.

◆

4.2. Desigualdades de conectividad

Las desigualdades de conectividad (3.4) son aquéllas que impiden que la solución se quede desconectada, es decir, que se formen subtours. Veamos bajo qué condiciones definen faceta del poliedro.

Teorema 4.2.1 *Si $S = \bigcup_{k \in Q} V_k$, donde $Q \subset \{1, \dots, p\}$, y $V \setminus S$ son conexos, entonces la desigualdad de conectividad*

$$\sum_{(i,j): i \in S, j \in V \setminus S} x_{ij} \geq 1$$

define una faceta de $\text{WGRP}(G)$.

Dem: Es evidente que la desigualdad de conectividad es de configuración. Su grafo de configuración G_C tiene dos vértices, B_1 y B_2 , y una sola arista no requerida (B_1, B_2) , por lo que $\dim(WGRP(G_C)) = 1$. Para ver que la desigualdad induce faceta de $WGRP(G_C)$ basta encontrar un tour que la cumpla con igualdad. Este tour será el que recorre la única arista del grafo configuración una vez en cada sentido, y que tiene coste 2, con lo que cumple con igualdad la restricción. Así pues, la desigualdad de conectividad define una faceta para el $WGRP(G_C)$. Aplicando el teorema 3.4.3, queda demostrado que también define una faceta del $WGRP(G)$. ♦

4.3. Desigualdades de corte R-impares

Aunque la formulación del WGRP no contiene explícitamente más desigualdades que las triviales, las de obligatoriedad y las de conectividad, hay una familia de desigualdades que aparecen de modo natural cuando relajamos la condición de integralidad de las variables x_{ij}, x_{ji} en (3.5): las *desigualdades de corte R-impares o de R-imparidad*.

Si las variables son enteras, las ecuaciones (3.3) del sistema, $\sum x_{ij} = \sum x_{ji}$ para todo vértice $i \in V$, hacen que el tour sea par, es decir, que $x(\delta(i)) = \sum_j x_{ij} + \sum_j x_{ji}$ sea par en cada vértice $i \in V$. Obviamente, si las variables no son enteras, esto no tiene por qué ocurrir.

Notemos que las desigualdades de obligatoriedad (3.2) implican que para cada vértice $i \in V$ se cumple que $x(\delta(i)) \geq |\delta_R(i)|$. Si el término $|\delta_R(i)|$ es par, no podemos deducir nada. Pero si $|\delta_R(i)|$ es *impar*, como cada tour para el WGRP tiene que ser par, necesariamente se ha de cumplir que

$$x(\delta(i)) \geq |\delta_R(i)| + 1$$

y ésta es una nueva desigualdad válida para el $WGRP(G)$ que llamaremos de *R-imparidad*. Para presentarlas de un modo más general, notemos que $\delta(i)$ es una cortadura, y el razonamiento anterior puede hacerse para cortaduras. Dada una cortadura de aristas $\delta(S)$, diremos que esta cortadura es *R-impar* si el número de aristas requeridas que contiene $\delta_R(S)$ es impar. Sabemos que el número de veces que un tour debe pasar por aristas de una cortadura ha de ser siempre par, ya que, como un tour es un ciclo cerrado, por cada vez que crucemos de un subconjunto a otro, deberemos cruzar otra vez de vuelta. De otro modo, no se podría cerrar el ciclo. Así pues, si en una cortadura el número de aristas requeridas es impar, es seguro que deberemos pasar por las aristas de dicha cortadura tantas veces como aristas requeridas haya, y, al menos, una vez más. Esta condición expresada en forma de desigualdad es:

$$x(\delta(S)) \geq |\delta_R(S)| + 1, \quad \forall S \text{ tal que } S \text{ es una cortadura R-impar} \quad (4.4)$$

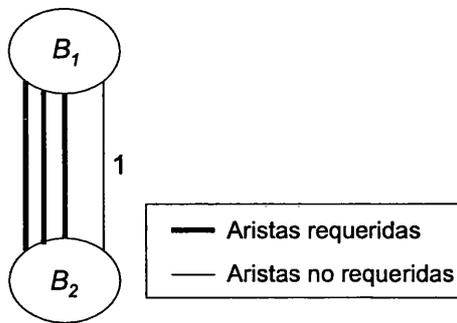


Figura 4.1: Grafo de configuración. Desigualdades de corte R-impar

Llamaremos *desigualdades de R-imparidad* a estas desigualdades, que son válidas para el WGRP(G). Veamos ahora que, bajo ciertas condiciones, inducen faceta del poliedro.

Teorema 4.3.1 Sea $S \subset V$ un subconjunto de nodos tal que S y $V \setminus S$ son conexos y $\delta(S)$ es una cortadura R-impar. Entonces, la desigualdad

$$x(\delta(S)) \geq |\delta_R(S)| + 1$$

define una faceta de WGRP(G).

Dem: La desigualdad de R-imparidad es de configuración. Veamos que esta desigualdad define una faceta para el WGRP en el grafo configuración asociado. Este grafo tiene dos nodos B_1 y B_2 , un número impar $|\delta(S)|$ de aristas requeridas y, posiblemente aunque no necesariamente, una arista no requerida (figura 4.1).

La dimensión del poliedro del WGRP en G_C es $\dim(WGRP(G_C)) = 2|E^C| - 1$. Escogemos una arista requerida e_0 , y el resto de aristas requeridas las dividimos en dos subconjuntos E_1 y E_2 de la misma cardinalidad. Construimos un vector base x^0 de la siguiente forma, donde \vec{x}_{ij} (\overleftarrow{x}_{ij}) representa a la componente correspondiente a recorrer la arista (i, j) de B_1 a B_2 (de B_2 a B_1):

$$\vec{x}_{ij} = \begin{cases} 1 & \text{si } (i, j) \in E_1 \\ 0 & \text{si } (i, j) \in E_2 \end{cases} \quad \overleftarrow{x}_{ij} = \begin{cases} 0 & \text{si } (i, j) \in E_1 \\ 1 & \text{si } (i, j) \in E_2 \end{cases}$$

Este es el vector que representa al grafo en el que las aristas de E_1 están dirigidas de B_1 a B_2 y las de E_2 en sentido contrario. Ahora, construimos los siguientes tours:

- Para cada arista $e \neq e_0$, creamos dos tours (podemos ver un ejemplo en la figura 4.2). El primero con:

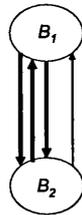


Figura 4.2: Ejemplo de WGRP-tour. Cortes R-impares

- $\vec{x}_e = \vec{x}_e^0 + 1$.
- $\overleftarrow{x}_{e_0} = \overleftarrow{x}_{e_0}^0 + 1$.
- $\vec{x}_e = \vec{x}_e^0$ y $\overleftarrow{x}_e = \overleftarrow{x}_e^0$ en otro caso.

y el segundo con:

- $\overleftarrow{x}_e = \overleftarrow{x}_e^0 + 1$.
- $\vec{x}_{e_0} = \vec{x}_{e_0}^0 + 1$.
- $\vec{x}_e = \vec{x}_e^0$ y $\overleftarrow{x}_e = \overleftarrow{x}_e^0$ en otro caso.

- Construimos, también, un tour que recorra la arista e_0 una vez en cada dirección:

- $\vec{x}_{e_0} = \vec{x}_{e_0}^0 + 1$.
- $\overleftarrow{x}_{e_0} = \overleftarrow{x}_{e_0}^0 + 1$.
- $\vec{x}_e = \vec{x}_e^0$ y $\overleftarrow{x}_e = \overleftarrow{x}_e^0$ en otro caso.

Si restamos el vector x^0 al resto de tours, y expresamos estos en forma matricial, ordenando las columnas y las filas de forma adecuada, queda la siguiente matriz:

$$\begin{pmatrix} E^C \setminus \{e_0\} & e_0 \\ \hline \text{I} & * \\ \hline 0 \dots 0 & 1 \ 1 \end{pmatrix}$$

Así puede observarse claramente que todos estos vectores son linealmente independientes, con lo que los tours que hemos construidos serán afinmente independientes. Podemos concluir, por tanto, que la desigualdad de R-imparidad define una faceta de $\text{WGRP}(G_C)$ y, aplicando el teorema 3.4.3, de $\text{WGRP}(G)$.

◆

Capítulo 5

Descripción del poliedro: Facetas derivadas de otros poliedros conocidos

En este capítulo estudiaremos otras familias de desigualdades válidas para el WGRP que definen facetas de su poliedro asociado. Estas desigualdades son similares a otras asociadas al poliedro del GRP definido sobre grafos no dirigidos y mixtos.

5.1. Desigualdades K-C estándar

Las desigualdades K-C (estándar) fueron introducidas por Corberán y Sanchis (1994) para el Problema del Cartero Rural. Discutiremos en esta sección su generalización al Windy General Routing Problem.

Consideraremos una partición del conjunto de nodos V en $K+1$ subconjuntos $\{M_0, M_1, \dots, M_{K-1}, M_K\}$, donde $K \geq 2$, cumpliendo las siguientes condiciones

- Cada subconjunto R-conexo (R-set) V_i está completamente contenido en $M_0 \cup M_K$ o en alguno de los subconjuntos M_j , $j = 1, \dots, K - 1$.
- Los subgrafos $G(M_i)$, $i = 0, \dots, K$ son conexos.
- Existe un número par (al menos 2) de aristas requeridas entre M_0 y M_K .

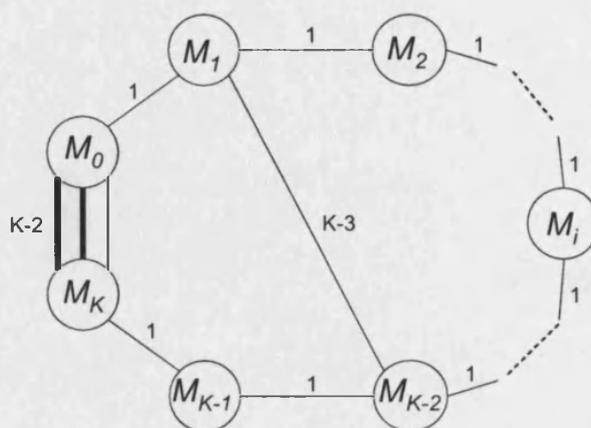


Figura 5.1: Grafo de configuración asociado a las desigualdades K-C

- Existe, al menos, una arista no requerida entre M_i y M_{i+1} , $i = 0, \dots, K-1$

Diremos que dos conjuntos M_i , M_j son consecutivos si $j = i + 1$.

Los coeficientes c los definimos de la siguiente manera:

- $c(M_0, M_K) = K - 2$
- $c(M_i, M_j) = |i - j|$ si $(i, j) \neq (0, K)$
- $c_0 = (K - 2)|\{(M_0 : M_K)_R\}| + 2(K - 1)$

De esta manera, la desigualdad K-C estándar sería:

$$\sum_{(i,j) \in E} (c(i, j)x_{ij} + c(j, i)x_{ji}) \geq c_0 \quad (5.1)$$

Es fácil ver que el tour del WGRP de menor coste en este grafo tiene coste c_0 . Cualquier tour debe pasar por todas las aristas requeridas al menos una vez, lo cual supone un coste de $(K - 2)|\{(M_0 : M_K)_R\}|$. Además, el tour debe visitar todos los M_i , partiendo y acabando en M_0 (o M_K). Este recorrido puede realizarse con un coste de $2(K - 1)$, por ejemplo visitando los nodos en el orden $M_0, M_1, \dots, M_{K-1}, M_0$ o $M_0, M_1, \dots, M_{K-2}, M_{K-1}, M_{K-2}, \dots, M_1, M_0$ (Figura 5.2), pero nunca con menor coste. Así pues, las desigualdades (5.1) son válidas para el WGRP.

Teorema 5.1.1 Sea $F(x) \geq c_0$ una desigualdad K-C estándar (5.1) y C su configuración asociada. Esta desigualdad define una faceta de $WGRP(G)$.

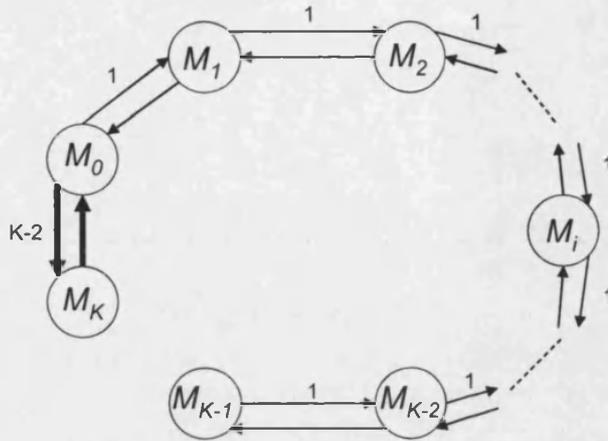


Figura 5.2: Tour de coste mínimo en el grafo de configuración K-C

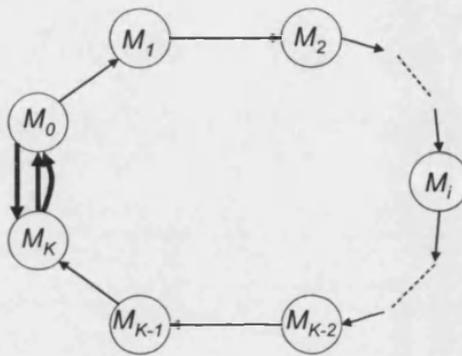


Figura 5.3: WGRP-tour 1 (K-C estándar)

Dem: La dimensión del poliedro $WGRP(G_C)$ es $2|E_C| - K$.

Orientemos la mitad de las aristas requeridas entre M_0 y M_K de M_0 a M_K , y la otra mitad en sentido contrario, y sea x^0 el vector de $\mathbb{R}^{2|E|}$ que representa dicho grafo dirigido. De esta manera, construiremos los tours necesarios añadiendo a x^1 aristas recorridas como sigue:

- Para cada arista $e \in (M_0 : M_K)$, construimos un tour que recorra e de M_0 a M_K y el camino entre M_0 y M_K formado por las aristas (M_i, M_{i+1}) , $i \in \{0, \dots, K-1\}$, de M_K a M_0 . La figura 5.3 muestra un tour de este tipo. Construimos, así mismo, otro tour recorriendo e de M_K a M_0 y las aristas de dicho camino de M_0 a M_K . En total, obtenemos así $2|(M_0 : M_K)|$ tours.
- Para cada arista $e = (M_i, M_{i+1})$, construimos un tour que recorra una vez en cada sentido las aristas del camino de aristas (M_j, M_{j+1}) de M_0 a M_K

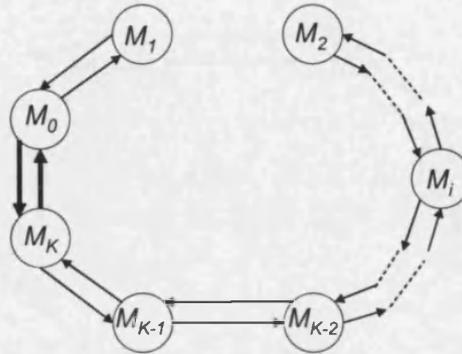


Figura 5.4: WGRP-tour 2 (K-C estándar)

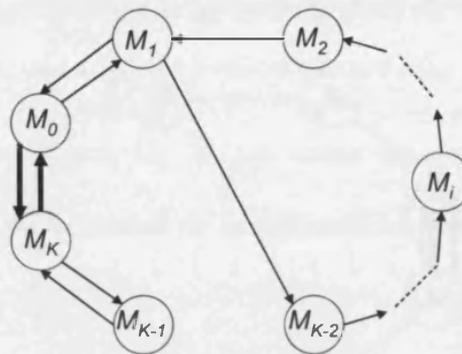


Figura 5.5: WGRP-tour 3 (K-C estándar)

excepto e . Así, obtenemos K tours como el de la figura 5.4.

- Para cada arista entre dos nodos no consecutivos $e = (M_i, M_j)$, $j \neq i + 1$, $i < j$, construimos. El camino de aristas entre nodos consecutivos entre M_i y M_j se recorre de M_j a M_i . El resto de aristas entre nodos consecutivos, excepto una, se recorre una vez en cada sentido. Invertiendo el sentido de recorrido de las aristas así descrito, obtenemos otro tour. Así, hemos conseguido $2 \sum_{i < j, j \neq i+1} |(M_i : M_j)|$ tours como el que puede verse en la figura 5.5.

En total, tenemos $2|E^c| - K$ tours que cumplen la restricción con igualdad. Si les restamos el vector x^0 y los expresamos en forma de matriz, ordenando filas y columnas adecuadamente, obtenemos:

$$\begin{array}{c}
 (M_0 : M_K) \quad (M_i : M_j) \quad (M_i : M_{i+1}) \\
 \left(\begin{array}{ccc|ccc}
 \text{I} & 0 & * & & & \\
 \hline
 0 & \text{I} & * & & & \\
 \hline
 0 & 0 & & 00 & 11 & \dots & 11 \\
 & & & 11 & 00 & \dots & 11 \\
 & & & \vdots & & \ddots & \vdots \\
 & & & 11 & 11 & \dots & 00
 \end{array} \right)
 \end{array}$$

Es fácil ver que la submatriz inferior derecha es de rango completo, con lo que lo es toda la matriz y los vectores son linealmente independientes. Así pues, la desigualdad K-C define faceta de $\text{WGRP}(G_C)$ y, aplicando el teorema 3.4.3, de $\text{WGRP}(G)$.



5.2. Desigualdades K-C₀₂

Las desigualdades K-C₀₂ fueron introducidas por Corberán, Romero y Sanchis (2003) para el Problema General de Rutas en un grafo mixto (MGRP). En esta sección, presentaremos su generalización al WGRP.

El grafo configuración correspondiente a estas desigualdades es el mismo que para las K-C, y puede observarse en la figura 5.6. La diferencia estriba en los costes de las aristas (es decir, los coeficientes de las variables en la desigualdad), que son los siguientes:

- $c(M_0, M_K) = K - 1$
- $c(M_0, M_1) = 0$
- $c(M_1, M_0) = 2$
- $c(M_i, M_j)$ toma como valor el coste del camino desde M_i hasta M_j usando las aristas $(M_i, M_{i+1}), \dots, (M_{j-1}, M_j)$ (suponiendo $i < j$), si $(i, j) \notin \{(0, K), (0, 1), (1, 0)\}$, es decir:

- $(A : V_1^i) \neq \emptyset$ y $(V_{n_i}^i : Z) \neq \emptyset$ para $i \in \{1, \dots, p\}$

Entonces, la desigualdad Path-Bridge, o PB, asociada a esta configuración tiene los siguientes coeficientes:

- $c(V_j^i, V_k^i) = \frac{|j-k|}{n_i-1} \quad \forall j, k \in \{0, 1, \dots, n_{i+1}\} \quad (V_j^i, V_k^i) \neq (V_0^i, V_{n_{i+1}}^i)$
- $c(A, Z) = 1$
- $c(V_j^i, V_k^r) = \frac{1}{n_i-1} + \frac{1}{n_r-1} + \left| \frac{j-1}{n_i-1} - \frac{k-1}{n_r-1} \right|, \quad \forall i, r \in \{1, \dots, p\}, i \neq r,$
 $j \in \{1, \dots, n_i\}, \quad k \in \{1, \dots, n_r\}$
- $c_0 = 1 + |(A : Z)_R| + \sum_{i=1}^p \frac{n_i + 1}{n_i - 1}$

donde, por comodidad de notación, identificamos A con V_0^i y Z con $V_{n_{i+1}}^i$.

Entonces, la desigualdad

$$\sum_{(i,j) \in E} (c(i,j)x_{ij} + c(j,i)x_{ji}) \geq c_0 \quad (5.2)$$

asociada a esta configuración es conocida como desigualdad PB.

Las aristas entre los nodos A y Z constituyen el *puente* (bridge), mientras que las aristas (V_j^i, V_{j+1}^i) que unen los nodos $A = V_1^i, V_2^i, \dots, V_{n_i}^i, V_{n_{i+1}}^i = Z$ forman los *caminos* (path), de ahí el nombre de estas desigualdades. Las desigualdades K-C son un caso particular de las Path-Bridge; en concreto, son desigualdades PB con un único camino ($p = 1$).

Puesto que se ha demostrado que las desigualdades PB son válidas para el GRP no dirigido (Letchford, 1997), también son válidas para WGRP(G). La figura 5.7 muestra un tour de coste mínimo para el WGRP definido sobre el grafo de configuración asociado a las desigualdades PB. Como la suma del número de aristas requeridas y el de "caminos" de A a Z por nodos V_i^j es impar, hemos tenido que recorrer dos veces una de las aristas requeridas para poder cerrar el tour. Esto tiene menor coste que cerrarlo volviendo a recorrer uno de los citados caminos. De esta manera, el coste total de recorrer las aristas requeridas es $|(A : Z)_R| + 1$. Además, el coste de recorrer cada camino i es $\frac{n_i+1}{n_i-1}$, con lo

que el coste total del tour es $c_0 = 1 + |(A : Z)_R| + \sum_{i=1}^p \frac{n_i + 1}{n_i - 1}$.

Teorema 5.3.1 Sea $F(x) \geq c_0$ una desigualdad Path-Bridge (5.2) y \mathcal{C} su configuración asociada. Esta desigualdad define una faceta de WGRP(G).

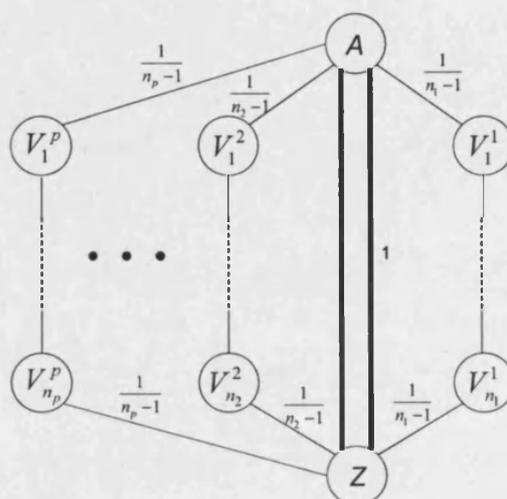


Figura 5.7: Grafo de configuración. Desigualdades Path-Bridge

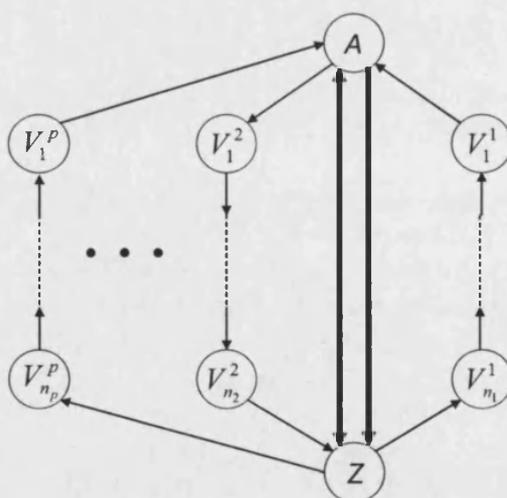
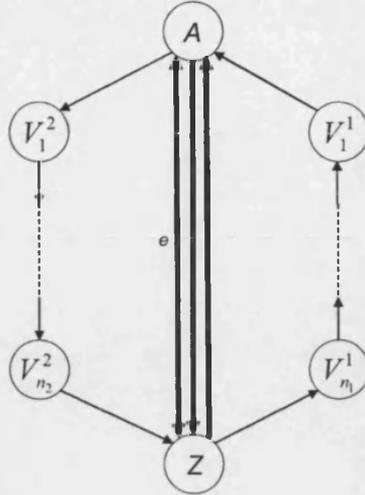


Figura 5.8: Tour de coste mínimo en el grafo de configuración Path Bridge

Figura 5.9: Tour x^1 (Path-Bridge)

Dem: Para la demostración distinguiremos dos casos, cuando el número de aristas requeridas b entre A y Z es par y cuando es impar.

Supongamos que b es par y, por tanto, el número de caminos p de A a Z es impar. A partir del grafo de configuración, podemos construir una instancia del GATSP. Para ello, desdoblamos todas las aristas en dos arcos opuestos, y los consideramos todos como no requeridos. Para el GATSP existe una familia de desigualdades válidas, llamadas desigualdades s -Path, que definen faceta del poliedro (Chopra y Rinaldi, 1996). Los coeficientes de dichas desigualdades son los mismos que para las Path-Bridge en el caso que nos ocupa, y el término de la derecha es $1 + \sum_{i=1}^p \frac{n_i+1}{n_i-1}$. Como dichas desigualdades definen facetas del poliedro del GATSP, que tiene dimensión $\dim(GATSP) = |A_{GATSP}| - |V_{GATSP}| + 1 = 2|E^C| - |V^C| + 1$, existirá dicho número de tours afínmente independientes cumpliendo la restricción con igualdad. Si ahora construimos un vector que represente a recorrer la mitad de las aristas requeridas entre A y Z en sentido de A a Z y la otra mitad en sentido opuesto, podemos sumar dicho vector a los tours del GATSP. El resultado será un conjunto de $2|E^C| - |V^C| + 1$ tours del WGRP en G_C que cumplen con igualdad la restricción Path-Bridge asociada. Además, seguirán siendo afínmente independientes, con lo que la desigualdad define una faceta de $WGRP(G_C)$ y, aplicando el teorema 3.4.3, de $WGRP(G)$.

Veamos ahora el caso en que el número de aristas requeridas b es impar y el número de caminos p es par. Elegimos una arista requerida e entre A y Z y construimos el siguiente tour x^1 . Recorremos la arista e una vez en cada sentido. Del resto de aristas requeridas entre A y Z recorremos la mitad de A a Z y la otra mitad de Z a A . Finalmente, recorremos la mitad de los p caminos en sentido de A a Z y la otra mitad en sentido opuesto (figura 5.9). Ahora, construimos los siguientes tours:

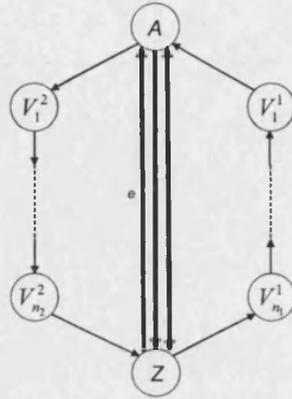


Figura 5.10: WGRP-tour 1 (Path-Bridge)

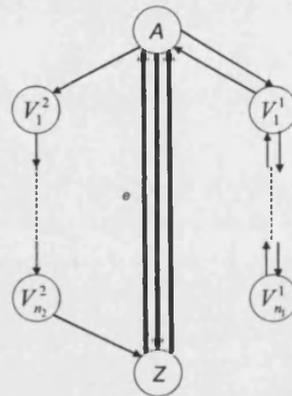


Figura 5.11: WGRP-tour 2 (Path-Bridge)

- Para cada arista de $(A : Z) \setminus \{e\}$, requerida o no, recorremos dicha arista en sentido de A a Z una vez más que el tour x_1 , y la arista e sólo en sentido de Z a A . El resto de las aristas se recorren como en x^1 (ver figura 5.10). Análogamente construimos otro tour que recorra la mencionada arista de Z a A , y la arista e de A a Z . De esta manera, conseguimos $2|(A : Z)| - 2$ tours.
- Para cada arista entre nodos consecutivos V_i^k, V_{i+1}^k , construimos un tour que no recorra dicha arista en ningún sentido. Además, recorremos todas las demás aristas del camino k en ambos sentidos. Si el camino k estaba orientado de A a Z en el tour x^1 , recorremos la arista e sólo una vez, de A a Z . De lo contrario, lo hacemos de Z a A . El resto de aristas se recorren como en x^1 (Figura 5.11). Así obtenemos $\sum_{i=1}^p (n_i + 1)$ tours.
- Para cada camino k , escogemos entre los tours construidos en el punto an-

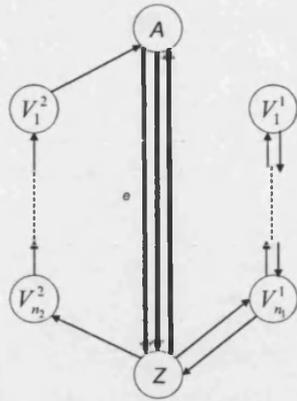


Figura 5.12: WGRP-tour 3 (Path-Bridge)

terior, aquél que no recorre la arista (A, V_i^k) . A partir de éste, construimos un nuevo tour invirtiendo el sentido en que se recorren todos los caminos distintos de k , así como la arista e . Así obtenemos p tours como el de la figura 5.12.

- Para cada arista entre dos nodos no consecutivos de un mismo camino V_i^k, V_j^k , construimos un tour que recorra dicha arista de V_i^k a V_j^k (supondremos $i < j$), y las aristas (V_i^k, V_{l+1}^k) , $i \leq l < j$ en sentido de V_{l+1}^k a V_i^k . El resto de aristas de ese camino las recorreremos en ambos sentidos, excepto la arista (V_{i-1}^k, V_i^k) (o la (V_j^k, V_{j+1}^k) si $i = 0$), que no la recorreremos. Si el camino k estaba orientado de A a Z en el tour x_1 , recorreremos la arista e sólo una vez, de A a Z . De lo contrario, lo hacemos de Z a A , como en el tour mostrado en la figura 5.13. El resto de aristas tienen la misma orientación que en x_1 . Consideraremos, también, los tours resultantes de invertir el sentido de las aristas del camino k . De esta manera, si E_1 es el conjunto de aristas entre nodos no consecutivos de un mismo camino, hemos obtenido $2|E_1|$ tours.
- Finalmente, por cada arista entre nodos V_i^k, V_j^l de caminos distintos, construimos un tour recorriendo dicha arista de V_j^l a V_i^k . Si $\frac{i-1}{n_k-1} > \frac{j-1}{n_l-1}$, recorreremos las aristas del camino k entre V_i^k y A de V_i^k a A y las del camino l entre Z y V_j^l en sentido de Z a A (el caso contrario sería análogo). El resto de aristas de estos caminos excepto dos, una de cada camino, las recorreremos en ambos sentidos. Escogeremos, además, la orientación de la arista e , así como la de los demás caminos, de manera que, recorriendo el resto de aristas entre A y Z como en x_1 , el resultado sea un tour para el WGRP en G_C . La figura 5.14 muestra un tour de este tipo. De la misma forma, podemos construir un tour que recorra la arista V_i^k, V_j^l en sentido opuesto. Si denotamos E_2 al conjunto de aristas entre nodos de camino distintos, hemos construido $2|E_2|$ tours.

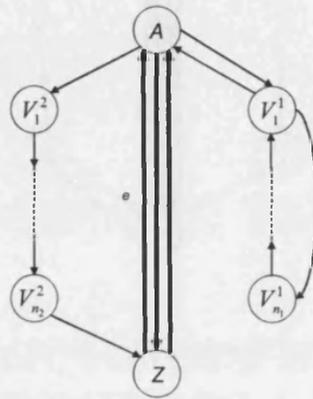


Figura 5.13: WGRP-tour 4 (Path-Bridge)

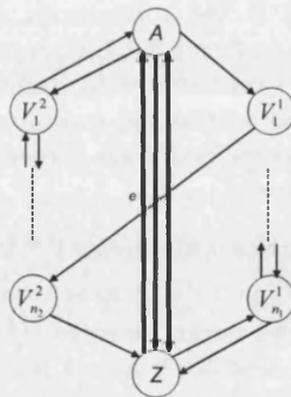


Figura 5.14: WGRP-tour 5 (Path-Bridge)

En total, contando el tour x^1 , tenemos $1 + 2|(A : Z)| - 2 + \sum_{i=1}^p (n_i + 1) + p + 2|E_1| + 2|E_2|$ tours, y todos cumplen la restricción Path-Bridge con igualdad. Por lo tanto, tenemos $2|E^C| - \sum_{i=1}^p n_i - 1$ tours. Veamos que son afinmente independientes. Para ello, restamos el vector x^1 a todos los demás y los representamos matricialmente:

$$\begin{array}{c} e \quad (A : Z) \setminus \{e\} \quad (V_i^k : V_{i+1}^k) \quad (V_i^k : V_j^k) \quad (V_i^k : V_j^l) \\ \left(\begin{array}{c|c|c|c|c} A & I & 0 & 0 & 0 \\ \hline * & 0 & B & 0 & 0 \\ \hline A & 0 & * & I & 0 \\ \hline * & * & * & * & I \end{array} \right) \end{array}$$

donde las matrices A y B son:

$$A = \begin{pmatrix} -1 & 0 \\ \vdots & \vdots \\ -1 & 0 \\ 0 & -1 \\ \vdots & \vdots \\ 0 & -1 \end{pmatrix}$$

$$B = \begin{pmatrix} B_1 & \begin{pmatrix} -1, 1 \\ 0 \end{pmatrix} & \dots & \begin{pmatrix} -1, 1 \\ 0 \end{pmatrix} & \begin{pmatrix} 1, -1 \\ 0 \end{pmatrix} & \begin{pmatrix} 1, -1 \\ 0 \end{pmatrix} & \dots & \begin{pmatrix} 1, -1 \\ 0 \end{pmatrix} \\ \begin{pmatrix} -1, 1 \\ 0 \end{pmatrix} & B_1 & \dots & \begin{pmatrix} -1, 1 \\ 0 \end{pmatrix} & \begin{pmatrix} 1, -1 \\ 0 \end{pmatrix} & \begin{pmatrix} 1, -1 \\ 0 \end{pmatrix} & \dots & \begin{pmatrix} 1, -1 \\ 0 \end{pmatrix} \\ \vdots & & \ddots & & & & & \vdots \\ \begin{pmatrix} -1, 1 \\ 0 \end{pmatrix} & \begin{pmatrix} -1, 1 \\ 0 \end{pmatrix} & \dots & B_1 & \begin{pmatrix} 1, -1 \\ 0 \end{pmatrix} & \begin{pmatrix} 1, -1 \\ 0 \end{pmatrix} & \dots & \begin{pmatrix} 1, -1 \\ 0 \end{pmatrix} \\ \begin{pmatrix} -1, 1 \\ 0 \end{pmatrix} & \begin{pmatrix} -1, 1 \\ 0 \end{pmatrix} & \dots & \begin{pmatrix} 1, -1 \\ 0 \end{pmatrix} & B_2 & \begin{pmatrix} 1, -1 \\ 0 \end{pmatrix} & \dots & \begin{pmatrix} 1, -1 \\ 0 \end{pmatrix} \\ \begin{pmatrix} -1, 1 \\ 0 \end{pmatrix} & \begin{pmatrix} -1, 1 \\ 0 \end{pmatrix} & \dots & \begin{pmatrix} 1, -1 \\ 0 \end{pmatrix} & \begin{pmatrix} 1, -1 \\ 0 \end{pmatrix} & B_2 & \dots & \begin{pmatrix} 1, -1 \\ 0 \end{pmatrix} \\ \vdots & & & \vdots & \vdots & & \ddots & \vdots \\ \begin{pmatrix} -1, 1 \\ 0 \end{pmatrix} & \begin{pmatrix} -1, 1 \\ 0 \end{pmatrix} & \dots & \begin{pmatrix} 1, -1 \\ 0 \end{pmatrix} & \begin{pmatrix} 1, -1 \\ 0 \end{pmatrix} & \begin{pmatrix} 1, -1 \\ 0 \end{pmatrix} & \dots & B_2 \end{pmatrix}$$

siendo (i, j) el vector horizontal cuyos elementos impares son todos i y los pares j . De esta manera, las submatrices del tipo

$$\begin{pmatrix} (i, j) \\ 0 \end{pmatrix}$$

tienen en su primera fila dicho vector y el resto de sus elementos son todos 0. La submatriz B_1 es la siguiente:

$$B_1 = \begin{pmatrix} -1 & 0 & 0 & 1 & 0 & 1 & \dots & 0 & 1 & 0 & 1 \\ 0 & 1 & -1 & 0 & 0 & 1 & \dots & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & -1 & 0 & \dots & 0 & 1 & 0 & 1 \\ \vdots & & & & & & & & & & \vdots \\ 0 & 1 & 0 & 1 & 0 & 1 & \dots & 0 & 1 & -1 & 0 \end{pmatrix}$$

es decir, una matriz en la que los elementos de la fila i y las columnas $2i - 1$ y $2i$ son -1 y 0 , respectivamente, mientras que el resto de elementos son 0 si la columna es impar y 1 si es par. La submatriz B_2 es igual a la B_1 , sólo que intercambian las columnas pares con las impares. Por lo tanto, si intercambiamos las columnas de las submatrices B_2 , obtenemos la siguiente matriz:

$$B = \begin{pmatrix} B_1 & (-1,1) & \dots & (-1,1) \\ & 0 & & 0 \\ (-1,1) & B_1 & \dots & (-1,1) \\ & 0 & & 0 \\ \vdots & & \ddots & \vdots \\ (-1,1) & (-1,1) & \dots & B_1 \\ & 0 & & 0 \end{pmatrix}$$

Para ver que la primera matriz es de rango completo, sólo necesitamos comprobar que B lo es. Supongamos que el vector nulo puede expresarse como combinación lineal de sus filas. Sea r el número de matrices B_1 que aparecen en B , y s el número de filas de B_1 . Sean α_{ij} , $1 \leq i \leq r$, $1 \leq j \leq s$ los coeficientes de la mencionada combinación lineal, siendo α_{ij} el coeficiente asociado a la j -ésima fila del bloque que contiene a la i -ésima matriz B_1 . Entonces, deben cumplirse las siguientes desigualdades:

$$-\sum_{i=1, i \neq k_1}^r \alpha_{i1} - \alpha_{k_1 l_1} = 0 \quad 1 \leq k_1 \leq r \quad 1 \leq l_1 \leq s$$

$$\sum_{i=1, i \neq k_2}^r \alpha_{i1} + \sum_{j=1, j \neq l_2}^s \alpha_{k_2 j} = 0 \quad 1 \leq k_2 \leq r \quad 1 \leq l_2 \leq s$$

Sumando los dos tipos de ecuaciones cuando $k_1 = k_2$ y $l_1 = l_2$, obtenemos las siguientes ecuaciones:

$$\sum_{j=1, j \neq l}^s \alpha_{kj} - \alpha_{kl} = 0 \quad 1 \leq k \leq r \quad 1 \leq l \leq s$$

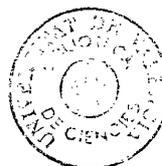
de donde se puede deducir fácilmente que $\alpha_{ij} = 0$ para todo par i, j . Por lo tanto, la matriz B es de rango completo.

Hemos llegado, pues, a que la matriz original es de rango completo, de donde se concluye que los tours construidos son afinmente independientes. Esto demuestra que la desigualdad define una faceta de $\text{WGRP}(G_C)$ y, aplicando el teorema 3.4.3, también de $\text{WGRP}(G)$.

◆

5.4. Desigualdades Path-Bridge₀₂

Vamos a considerar aquí las desigualdades Path-Bridge₀₂. Éstas fueron propuestas por Corberán, Romero y Sanchis (2003) para el MGRP.



El grafo de configuración de estas desigualdades es igual al de las PB estándar, sólo que ahora es suficiente con que el número de nodos en cada camino i , n_i , sea mayor o igual que 1. Consideraremos dos tipos de caminos en esta configuración, los caminos que llamaremos de tipo AZ y los que denominaremos de tipo ZA . Los costes en el grafo configuración se definen de la siguiente manera:

- $c(A, Z) = c(Z, A) = 1$.
- $c(V_0^i, V_1^i) = 0$ y $c(V_1^i, V_0^i) = \frac{2}{n_i}$ para cada camino i del tipo AZ .
- $c(V_0^i, V_1^i) = \frac{2}{n_i}$ y $c(V_1^i, V_0^i) = 0$ para cada camino i del tipo ZA .
- $c(V_j^i, V_{j+1}^i) = c(V_{j+1}^i, V_j^i) = \frac{1}{n_i}$.
- $c(V_j^i, V_k^i)$, para dos nodos V_j^i, V_k^i no consecutivos, toma el valor del camino más corto de V_j^i a V_k^i usando sólo aristas del camino i , o lo que es lo mismo:
 - $c(V_j^i, V_k^i) = \frac{|j-k|}{n_i} \quad \forall j, k \in \{1, 2, \dots, n_i + 1\}$
 - $c(V_0^i, V_k^i) = \frac{k-1}{n_i}$ y $c(V_k^i, V_0^i) = \frac{k+1}{n_i}$ si el camino i es del tipo AZ , $1 \leq k \leq n_i$.
 - $c(V_0^i, V_k^i) = \frac{k+1}{n_i}$ y $c(V_k^i, V_0^i) = \frac{k-1}{n_i}$ si el camino i es del tipo ZA , $1 \leq k \leq n_i$.
- $c(V_j^i, V_l^k)$, $i \neq k$, se calcula mediante un lifting secuencial. Para ello, sean $\{e_1, \dots, e_h\}$ las aristas de este tipo ordenadas de manera arbitraria. Para $i = 1$ hasta h , si $e_i = (u, v)$, asignamos a $c(u, v)$ el valor máximo de manera que exista un tour de coste $|(A : Z)_R| + p + 1$ que recorra e_i sólo de u a v y no pase por las aristas $\{e_{i+1}, \dots, e_h\}$. Así mismo, asignamos a $c(v, u)$ el valor máximo tal que exista un tour del mismo coste que recorra e_i de v a u (pudiendo también recorrerla, si fuera necesario, de u a v), pero que no pase por las aristas $\{e_{i+1}, \dots, e_h\}$.
- $c_0 = |(A : Z)_R| + p + 1$.

La principal diferencia entre estas desigualdades y las PB estándar es que la primera arista de cada camino tiene un coste en un sentido, $\frac{2}{n_i}$, que es el doble del coste del resto de aristas del camino, $\frac{1}{n_i}$, mientras que en el sentido opuesto tiene coste 0 (figura 5.15). Esta arista con costes diferentes al resto puede ser cualquiera de las que componen el camino. La dirección del arco con coste 0 determina si el camino correspondiente es del tipo AZ o ZA .

Es fácil ver que, cuando $p = 1$, las desigualdades PB_{02} son equivalentes a las desigualdades $K-C_{02}$.

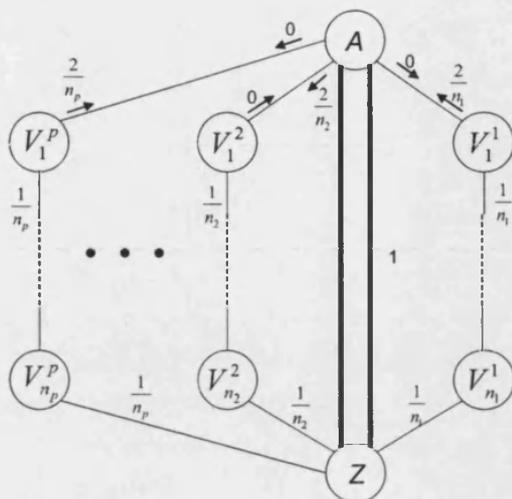


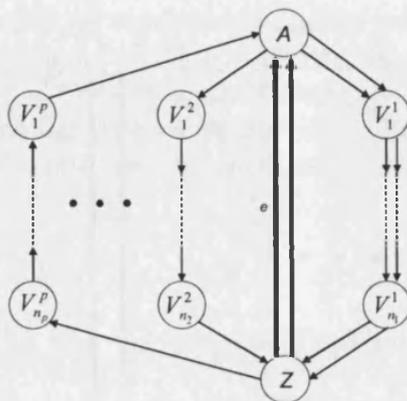
Figura 5.15: Grafo de configuración asociado a las desigualdades Path-Bridge₀₂

Una demostración similar a la de Corberán, Romero y Sanchis (2003) probaría que las PB₀₂ son válidas para el WGRP. Veamos a continuación que también tienen la propiedad de definir faceta de WGRP(G).

Teorema 5.4.1 Sea $F(x) \geq c_0$ una desigualdad Path-Bridge₀₂, y sean p_{AZ} y p_{ZA} el número de caminos de tipo AZ y $Z A$, respectivamente, que contiene el grafo de configuración asociado a dicha desigualdad. Si $|(A : Z)| \geq |p_{AZ} - p_{ZA}| + 1$, entonces la desigualdad define una faceta de WGRP(G).

Dem: Elegimos una arista requerida e entre A y Z y construimos el siguiente tour x^1 . Recorremos la arista e una vez en cada sentido. Del resto de aristas requeridas entre A y Z , si $p_{AZ} \geq p_{ZA}$ recorremos $p_{AZ} - p_{ZA}$ de Z a A . En caso contrario, recorremos $p_{ZA} - p_{AZ}$ de A a Z . De las restantes aristas requeridas entre A y Z , excepto e , recorremos la mitad de A a Z y la otra mitad de Z a A . Finalmente, recorremos los caminos del tipo AZ en sentido de A a Z y el resto en sentido opuesto. Ahora, construimos los siguientes tours:

- Para cada arista de $(A : Z) \setminus \{e\}$ recorremos dicha arista en sentido de A a Z una vez más que el tour x^1 , y la arista e sólo en sentido de Z a A . El resto de las aristas se recorren como en x^1 . Análogamente, construimos otro tour que recorra la mencionada arista de Z a A , y la arista e de A a Z . Estos tours son iguales a los construidos para las desigualdades Path-Bridge (figura 5.10), y suman en total $2|(A : Z)| - 2$ tours.
- Para cada arista entre nodos consecutivos V_i^k, V_{i+1}^k , construimos un tour que no recorra dicha arista en ningún sentido. Además, recorremos todas las demás aristas del camino k en ambos sentidos. Si el camino k es de

Figura 5.16: WGRP-tour 1 (Path-Bridge₀₂)

tipo AZ , recorreremos la arista e sólo una vez, de A a Z . De lo contrario, lo hacemos de Z a A . El resto de aristas se recorren como en x^1 . También estos tours tienen su equivalente para las Path-Bridge (figura 5.11). Así obtenemos $\sum_{i=1}^p (n_i + 1)$ tours.

- Para cada camino k , construimos un tour que recorre dicho camino dos veces en el mismo sentido, de A a Z si k es del tipo AZ , o en el contrario, si es del tipo ZA . La arista e se recorre sólo una vez, en el sentido opuesto que el camino k , y el resto de aristas se recorren como en x^1 (Figura 5.16). Así conseguimos otros k tours.
- Para cada par de nodos no consecutivos de un mismo camino V_i^k, V_j^k , construimos un tour que recorra dicha arista de V_i^k a V_j^k (supondremos $i < j$), y las aristas (V_l^k, V_{l+1}^k) , $i \leq l < j$ en sentido de V_{l+1}^k a V_l^k . El resto de aristas de ese camino las recorreremos en ambos sentidos, excepto la arista (V_{i-1}^k, V_i^k) (o la (V_j^k, V_{j+1}^k) si $i = 0$), que no la recorreremos. Si el camino k estaba orientado de A a Z en el tour x^1 , recorreremos la arista e sólo una vez, de A a Z . De lo contrario, lo hacemos de Z a A . El resto de aristas tienen la misma orientación que en x^1 . Consideraremos, también, los tours resultantes de invertir el sentido de las aristas del camino k . De esta manera, si E_1 es el conjunto de aristas entre nodos no consecutivos de un mismo camino, hemos obtenido $2|E_1|$ tours (ver figura 5.13).
- Por cada arista entre nodos V_i^k, V_j^l de caminos distintos, dada la definición de los costes asociados a dicha arista, podemos construir dos tours, cada uno recorriendo la arista en una dirección. Así tenemos $2|E_2|$ tours (ver figura 5.14), donde E_2 representa al conjunto de tales aristas.

Si a estos tours les restamos el tour x^1 y los expresamos en forma matricial, obtenemos:

$$\begin{array}{c}
 e \quad (A : Z) \setminus \{e\} \quad (V_i^k : V_{i+1}^k) \quad (V_i^k : V_j^k) \quad (V_i^k : V_j^l) \\
 \left(\begin{array}{c|c|c|c|c}
 A & I & 0 & 0 & 0 \\
 \hline
 * & 0 & B & 0 & 0 \\
 \hline
 A & 0 & * & I & 0 \\
 \hline
 * & * & * & * & C
 \end{array} \right)
 \end{array}$$

donde

$$A = \begin{pmatrix} -1 & 0 \\ \vdots & \vdots \\ -1 & 0 \\ 0 & -1 \\ \vdots & \vdots \\ 0 & -1 \end{pmatrix}$$

$$B = \begin{pmatrix} 10 & \dots & 10 & 00 & \dots & 00 & \dots & 00 & \dots & 00 \\
 00 & \dots & 00 & 10 & \dots & 10 & & 00 & \dots & 00 \\
 \vdots & & & & & & \ddots & & & \vdots \\
 00 & \dots & 00 & 00 & \dots & 00 & \dots & 10 & \dots & 10 \end{pmatrix}$$

que es de rango completo, y C es una matriz diagonal y, por tanto, de rango completo también. Así pues, los vectores son linealmente independientes, y todos los tours son afinmente independientes. Esto prueba que la desigualdad PB_{02} define una faceta de $WGRP(G_C)$ y, aplicando el teorema 3.4.3, de $WGRP(G)$.

Corolario 5.4.2 *Puesto que las desigualdades $K-C_{02}$ se obtienen a partir de las PB_{02} cuando $p = 1$, también definen faceta de $WGRP(G)$.*



5.5. Desigualdades Honeycomb

Estudiaremos en este apartado otra familia de desigualdades que generalizan a las K-C en un sentido diferente al de las PB.

Las desigualdades Honeycomb fueron propuestas originalmente para el GRP (no dirigido) por Corberán y Sanchis (1998) y, al igual que las desigualdades PB, son una generalización de las desigualdades K-C. Sin embargo, se trata de una generalización en una dirección distinta y ninguna clase contiene a la otra. En una configuración K-C, una componente R-conexa se divide en dos conjuntos. Aunque podríamos generalizar esta configuración tanto en el número de partes en que una componente R-conexa se divide como en el número de componentes a dividir, aquí nos limitaremos a presentar la primera de estas generalizaciones, es decir, aquella en la que una componente R-conexa se divide en cualquier número de partes. La razón es que, aunque se puede demostrar fácilmente que la configuración resultante de ambas generalizaciones proporciona una desigualdad válida para el WGRP, no resulta nada sencillo demostrar que también inducen faceta del poliedro asociado. Por tanto, nos centraremos únicamente en el subconjunto de estas desigualdades que cumple las siguientes condiciones:

Sea p el número de componentes R-conexas del grafo (supondremos que este número es mayor o igual a 3) y $B = \{B_1, B_2, \dots, B_L, B_{L+1}, \dots, B_K\}$ una partición del conjunto de nodos V , cumpliendo:

- $3 \leq K \leq p, 2 \leq L \leq K$
- Cada uno de los conjuntos $(B_1 \cup B_2 \cup \dots \cup B_L), B_{L+1}, \dots, B_K$ contiene completamente al menos un conjunto R-conexo V_i , y cada V_i está completamente contenido en alguno de dichos conjuntos. Esta partición B define un grafo de configuración $G^C = (B, E^C)$ con un conjunto de nodos B y un conjunto de aristas E^C formado por una arista requerida (B_i, B_j) por cada arista requerida $e \in (B_i : B_j)_R$ y una arista no requerida (B_i, B_j) entre cada par de nodos B_i, B_j tal que $(B_i : B_j)_{NR} \neq \emptyset$
- El número $\delta_R(B_i)$ de aristas requeridas incidentes con cada conjunto B_i , $i \in \{1, \dots, L\}$ es par.
- El subgrafo inducido en el grafo de configuración por las aristas requeridas incidentes con los nodos B_1, \dots, B_L es conexo.
- Existe un árbol generador T (Figura 5.18) en el grafo de configuración tal que el grado de B_i en T es 1 para $i \in \{1, \dots, L\}$ y al menos 2 para $i \in \{L+1, \dots, K\}$, y cualquier camino en T entre dos nodos B_i, B_j , $i, j \in \{1, \dots, L\}$ tiene longitud 3 o mayor.
- Los subgrafos $G(B_i)$ son conexos.

Definimos los coeficientes de la desigualdad de la siguiente manera:

- $c(B_i, B_j) = d(B_i, B_j) - 2$ si $i, j \in \{1, \dots, L\}$.

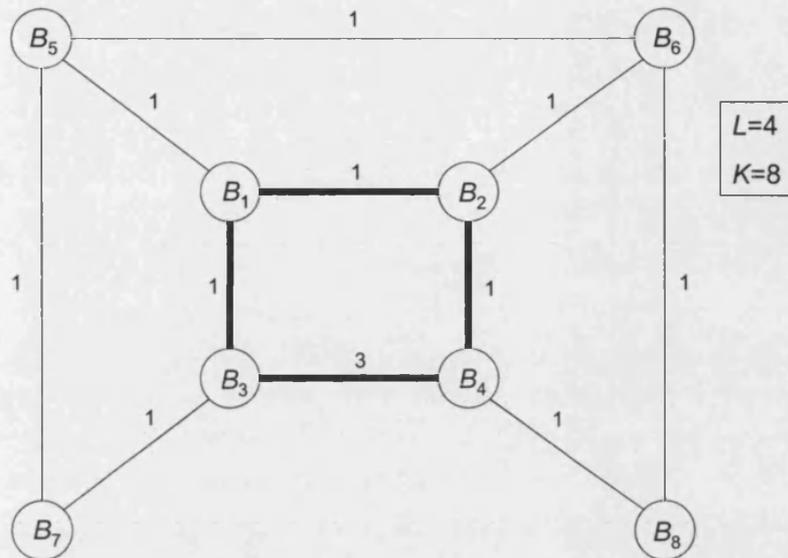


Figura 5.17: Grafo de configuración asociado a una desigualdad Honeycomb

- $c(B_i, B_j) = d(B_i, B_j)$ en el resto de casos.
- $c_0 = 2 + \sum_{(B_i, B_j) \in E^C} c(B_i, B_j)$

donde $d(B_i, B_j)$ es el número de aristas en el camino más corto desde B_i hasta B_j en el árbol T .

Un ejemplo de configuración Honeycomb puede verse en la Figura 5.17.

Dado que las desigualdades Honeycomb son válidas para el GRP no dirigido (Corberán y Sanchis, 1998), son también válidas para el WGRP. Demostraremos a continuación que también definen facetas de $WGRP(G)$.

Teorema 5.5.1 Sea $F(x) \geq c_0$ una desigualdad Honeycomb y C su configuración asociada. Esta desigualdad define una faceta de $WGRP(G)$.

Dem: La dimensión de $WGRP(G_C)$ es $2|E^C| - K + 1$. Denotamos $V_L = \{B_1, B_2, \dots, B_L\}$ y $V_K = \{B_{L+1}, B_{L+2}, \dots, B_K\}$. Comenzamos construyendo un ciclo que visite los nodos de V_L , pasando exactamente una vez por cada arista requerida y ninguna por las no requeridas. Como el grafo inducido por las aristas requeridas es conexo, y el grado de cada uno de los nodos de este grafo es par, podemos asegurar que tal ciclo existe. Llamaremos x^d al vector que representa a este ciclo. Sea T un árbol cumpliendo las condiciones mencionadas en la

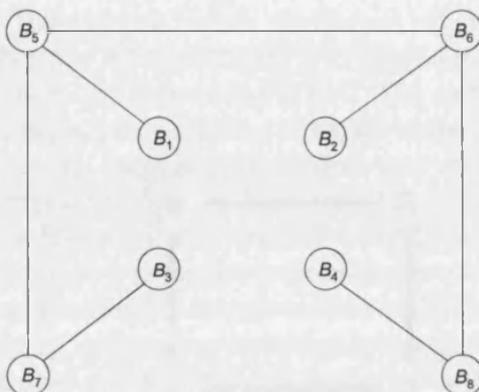


Figura 5.18: Árbol T asociado al grafo configuración de la figura 5.17

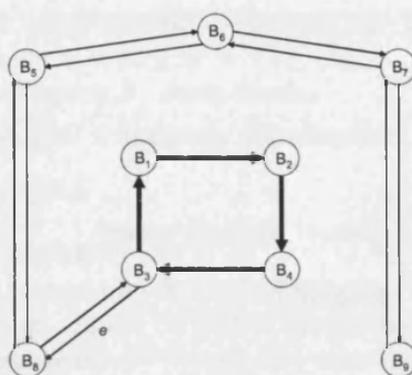


Figura 5.19: WGRP-tour 1 (Honeycomb)

descripción de la configuración Honeycomb. Denotaremos por T_1 al conjunto de aristas resultante de eliminar de T las aristas incidentes con los nodos de V_L , y $G(T_1)$ al subgrafo inducido por T_1 . A partir del vector x^d , construimos los siguientes tours:

- **Tipo 1** Por cada arista $e \in T \setminus T_1$, construimos un tour que recorra en ambos sentidos las aristas de T_1 y dicha arista. Así, obtenemos L tours como el mostrado en la figura 5.19.
- **Tipo 2** Sea $e_0 \in T \setminus T_1$. Por cada arista e entre B_i y B_j (no perteneciente a T), $i, j \in \{L+1, \dots, K\}$, construimos un tour que recorra dicha arista de B_i a B_j , las aristas del camino en T_1 entre B_j y B_i en sentido de B_j a B_i , y el resto de aristas de T_1 y la arista e_0 en ambos sentidos (Figura 5.20). También consideraremos los tours obtenidos al invertir el sentido de recorrido de todas las aristas, con lo que tenemos un total de $2(|E(V_K)| - (K - L - 1))$ tours.

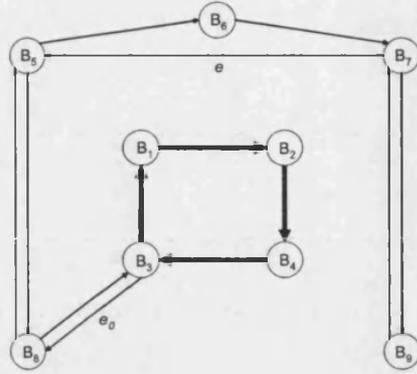


Figura 5.20: WGRP-tour 2 (Honeycomb)

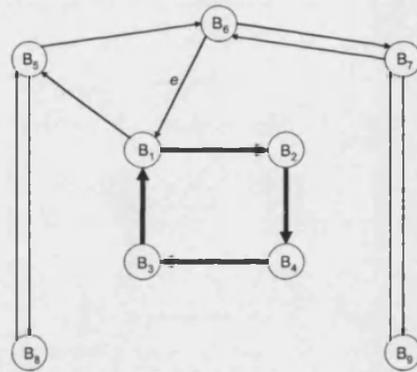


Figura 5.21: WGRP-tour 3 (Honeycomb)

- **Tipo 3** Por cada arista $e \in (V_L : V_K) \setminus T$ entre B_i y B_j , construimos un tour recorriéndola de B_i a B_j , y recorriendo el camino entre B_j y B_i en T de B_j a B_i . El resto de aristas de T_1 las recorreremos en ambos sentidos (Figura 5.21). Los tours resultantes de invertir el sentido de los así contruidos también serán tenidos en cuenta, con lo que conseguimos $2(|(V_L : V_K)| - L)$ tours.
- **Tipo 4** Por cada arista e de $G(V_L)$ entre B_i y B_j , construimos un tour recorriendo dicha arista de B_i a B_j y las aristas del camino de B_j a B_i en T en sentido de B_j a B_i . Las aristas restantes de T_1 se recorren en ambos sentidos (Figura 5.22). De esta manera hemos construido $2|E(V_L)|$ tours.
- **Tipo 5** Por cada arista $e_1 \in T_1$, sean G^1 y G^2 los dos subgrafos en que quedaría dividido $G(T_1)$ si elimináramos dicha arista. Escogemos una arista $e_2 \in T \setminus T_1$ entre $B_i \in V_L$ y $B_j \in V_{G^1}$. Análogamente, escogemos una arista $e_3 \in T \setminus T_1$ incidente con un nodo en V_{G^2} y otro de V_L . Ahora,

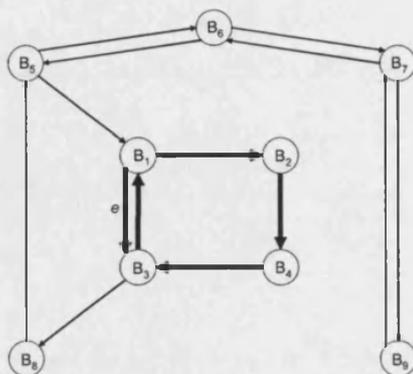


Figura 5.22: WGRP-tour 4 (Honeycomb)

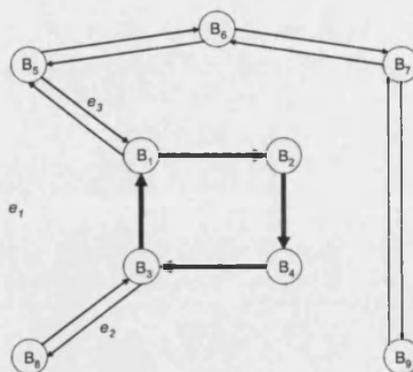


Figura 5.23: WGRP-tour 5 (Honeycomb)

recorremos en ambos sentidos todas las aristas de $T_1 \setminus \{e_1\} \cup \{e_2, e_3\}$ (Figura 5.23). Así hemos logrado obtener $K - L - 1$ tours.

En total, hemos construido $2|E^c| - K + 1$ tours que, como veremos a continuación, son linealmente independientes. Para probarlo, representamos los tours matricialmente, después de haberles restado el vector x^d , que, como se ha dicho al comienzo de la demostración, representa al ciclo que visita los nodos de V_L . Reordenando las filas, obtenemos la siguiente matriz, donde 1 representa una matriz con el valor 1 en todos sus elementos:

$$\begin{array}{l}
 \text{tipo5} \\
 \text{tipo1} \\
 \text{tipo2} \\
 \text{tipo3} \\
 \text{tipo4}
 \end{array}
 \left(\begin{array}{c|c|c|c|c}
 T \setminus T_1 & T_1 & E(V_K) \setminus T & (V_L : V_K) \setminus T & E(V_L) \\
 \hline
 * & A & 0 & 0 & 0 \\
 \hline
 B & 1 & 0 & 0 & 0 \\
 \hline
 * & * & I & 0 & 0 \\
 \hline
 * & * & 0 & I & 0 \\
 \hline
 * & * & 0 & 0 & I
 \end{array} \right)$$

Para probar que esta matriz es de rango completo basta con comprobar que

$$\left(\begin{array}{c|c}
 * & A \\
 \hline
 B & 1
 \end{array} \right)$$

lo es. Si sustituimos A y B por las correspondientes matrices, nos queda

$$\left(\begin{array}{c|c}
 a_{11} & a_{12} & \dots & a_{1r} & 00 & 11 & \dots & 11 \\
 a_{21} & a_{22} & \dots & a_{2r} & 11 & 00 & \dots & 11 \\
 \vdots & & & \vdots & \vdots & & & \vdots \\
 a_{s1} & a_{s2} & \dots & a_{sr} & 11 & 11 & \dots & 00 \\
 \hline
 11 & 00 & \dots & 00 & & & & \\
 00 & 11 & \dots & 00 & & & & \\
 \vdots & & & \vdots & & & & \\
 00 & 00 & \dots & 11 & & & & \\
 & & & & & & & 1
 \end{array} \right)$$

Supongamos que existe una combinación lineal de sus filas con coeficientes $\alpha_1, \dots, \alpha_s, \beta_1, \dots, \beta_{r/2}$ que da como resultado el vector nulo. Esto, expresado

en forma de ecuaciones, sería:

$$\sum_{i=1}^s \alpha_i a_{i1} + \beta_1 = 0 \quad (1)$$

$$\sum_{i=1}^s \alpha_i a_{i2} + \beta_1 = 0 \quad (2)$$

$$\sum_{i=1}^s \alpha_i a_{i3} + \beta_2 = 0 \quad (3)$$

⋮

$$\sum_{i=1}^s \alpha_i a_{ir} + \beta_{r/2} = 0 \quad (r)$$

$$\sum_{i=2}^s \alpha_i + \sum_{i=1}^{r/2} \beta_i = 0 \quad (r+1)$$

$$\sum_{i=1, i \neq 2}^s \alpha_i + \sum_{i=1}^{r/2} \beta_i = 0 \quad (r+2)$$

⋮

$$\sum_{i=1}^{s-1} \alpha_i + \sum_{i=1}^{r/2} \beta_i = 0 \quad (r+s)$$

Restando las ecuaciones (r+1) y (r+2) obtenemos $\alpha_1 = \alpha_2$. En general, restando (r+i) y (r+j) obtenemos $\alpha_i = \alpha_j$, de lo que se deduce $\alpha_1 = \alpha_2 = \dots = \alpha_s = -\frac{1}{s-1} \sum_{i=1}^{r/2} \beta_i$. Entonces, podemos reescribir las ecuaciones (1) hasta (r) como:

$$-\frac{1}{s-1} \sum_{i=1}^{r/2} \beta_i \sum_{j=1}^s a_{ij} + \beta_{\lfloor \frac{r+1}{2} \rfloor} = 0 \quad (j)$$

Si ahora sumamos las ecuaciones (1) hasta (r) con su nueva forma, obtenemos:

$$\begin{aligned} & -\frac{1}{s-1} \sum_{i=1}^{r/2} \beta_i \sum_{j=1}^s \sum_{k=1}^r a_{ijk} + 2 \sum_{i=1}^{r/2} \beta_i \\ & = \sum_{i=1}^{r/2} \beta_i \left(2 - \frac{1}{s-1} \sum_{j=1}^s \sum_{k=1}^r a_{ijk} \right) = 0 \end{aligned}$$

Si el término entre paréntesis valiera 0, tendríamos

$$\sum_{i=1}^s \sum_{j=1}^r a_{ij} = 2(s-1)$$

Por la estructura de los tours de tipo 5, sabemos que $\sum_{j=1}^r a_{ij} = 4$, por lo que tenemos $4s = 2(s-1)$ y $s = -1$, que es absurdo. Por tanto, $\sum_{i=1}^{r/2} \beta_i = 0$. Así pues, $\alpha_1 = \alpha_2 = \dots = \alpha_s = 0$. Sustituyendo los α por ceros en las ecuaciones (1) hasta (r), tenemos $\beta_1 = \beta_2 = \dots = \beta_{r/2} = 0$, con lo que queda demostrado que la matriz es de rango completo. De esta manera, hemos probado que los tours construidos son linealmente independientes, con lo que la desigualdad Honeycomb define una faceta de $\text{WGRP}(G_C)$. Aplicando el teorema 3.4.3, llegamos al resultado de que también define una faceta de $\text{WGRP}(G)$. ♦

5.6. Desigualdades Honeycomb₀₂

Al igual que las desigualdades K-C y Path-Bridge, las desigualdades Honeycomb tienen una versión alternativa en la que algunas de las aristas del grafo de configuración presentan coeficientes 0-2. Estas desigualdades fueron presentadas por Corberán, Mejía y Sanchis (2005aa) para el MGRP.

El grafo de configuración es el mismo que para las desigualdades Honeycomb estándar, para las que está definido por una partición de V en K subconjuntos $B_1, B_2, \dots, B_L, B_{L+1}, \dots, B_K$ cumpliendo una serie de condiciones descritas en la sección anterior. A su vez, existe un árbol generador T tal que el grado de los nodos B_i en T es 1 para $1 \leq i \leq L$ y mayor o igual a 2 para $L+1 \leq i \leq K$. Además, clasificaremos los nodos B_1, \dots, B_L en dos tipos, los de tipo \mathcal{O} , que serán los que tengan aristas incidentes con coeficientes 0-2, y los de tipo \mathcal{I} , el resto. Consideraremos que hay, al menos, un nodo de cada tipo. Entonces, si $d(B_i, B_j)$ es el número de aristas del camino más corto en T entre B_i y B_j , ha de cumplirse:

- $d(B_i, B_j) \geq 3$ si $1 \leq i, j \leq L$ y B_i y B_j son del mismo tipo.
- $d(B_i, B_j) \geq 2$ si $1 \leq i, j \leq L$ y B_i y B_j son de distinto tipo.

Obsérvese que, para las desigualdades Honeycomb estándar, la longitud de estos caminos debía ser al menos 3 para todos los casos.

Los coeficientes de la desigualdad Honeycomb₀₂ son los siguientes:

- $c(B_i, B_j) = c(B_j, B_i) = d(B_i, B_j) - 2$ si $1 \leq i, j \leq L$ y B_i y B_j son del mismo tipo.

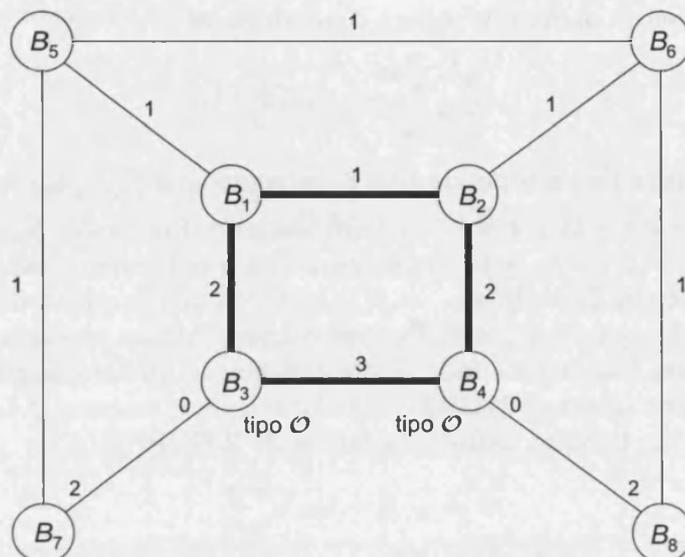


Figura 5.24: Grafo configuración asociado a las desigualdades Honeycomb₀₂

- $c(B_i, B_j) = c(B_j, B_i) = d(B_i, B_j) - 1$ si $1 \leq i, j \leq L$ y B_i y B_j son de distinto tipo.
- $c(B_i, B_j) = 0$ y $c(B_j, B_i) = 2$ si $1 \leq i \leq L$, $L + 1 \leq j \leq K$, $(B_i, B_j) \in T$ y B_i es de tipo \mathcal{O} .
- $c(B_i, B_j) = c(B_j, B_i) = 1$ si $(B_i, B_j) \in T$ y ninguno de los nodos es de tipo \mathcal{O} .
- $c(B_i, B_j) = c$ -coste del camino más corto en T de B_i a B_j en otro caso.
- $c_0 = 2(K - L) + \sum_{(B_i, B_j) \in E_R} c(B_i, B_j)$.

En la figura 5.24 podemos ver un ejemplo de configuración Honeycomb₀₂ con sus correspondientes coeficientes.

Teorema 5.6.1 Las desigualdades Honeycomb₀₂ $F(x) \geq c_0$ son válidas para el WGRP(G).

Dem: Probaremos que la desigualdad Honeycomb₀₂ es válida para el WGRP definido en G_C de lo que se concluye que será válida para el WGRP en G . Bastará probarlo para el caso en que G_C es un grafo completo.

Veamos que para un tour cualquiera x del WGRP en G_C se cumple $F(x) \geq c_0$. Podemos suponer que x sólo recorre aristas requeridas o aristas en T , ya que

cualquier otra arista que recorra puede ser sustituida por un camino de aristas requeridas y aristas en T con el mismo c -coste. Como x ha de recorrer al menos una vez cada arista requerida, tenemos ya un c -coste de $\sum_{(B_i, B_j) \in E_R} c(B_i, B_j)$. Además, el grafo inducido por las aristas requeridas es par y conexo, por lo que todo lo que añadamos ha de formar ciclos. Veamos que podemos suponer que estos ciclos están formados únicamente por aristas de T .

Si alguno de estos ciclos extra recorre de manera consecutiva dos aristas requeridas (B_i, B_j) y (B_j, B_k) , podemos sustituirlas por la arista (B_i, B_k) .

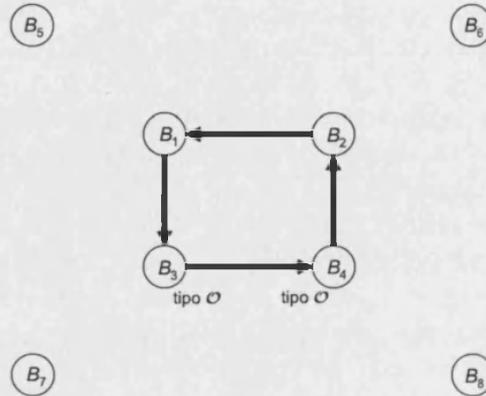
Sea e una arista requerida recorrida por algún ciclo extra, z el vector que representa a dicho ciclo, y l el número de aristas de T recorridas por z . Si el ciclo no contiene ninguna arista de T , podemos eliminarlo sin problemas. Supongamos que el ciclo contiene también aristas no requeridas de T . Si recorre una única arista con c -costes 0-2, podemos suponer que lo hace en el sentido de menor coste. En este caso, e tendrá un c -coste de $l - 1$, y el ciclo cumplirá $F(z) = 2(l - 1)$. Si añadimos una copia del ciclo en sentido opuesto, el c -coste aumenta en $2l$ unidades. Ahora, podemos eliminar los dos arcos opuestos correspondientes a recorrer la arista e en ambos sentidos, así como los otros dos arcos de una de las aristas adyacentes a e , sin que esto afecte a la validez del tour. Esto supone una disminución del c -coste de $2l$ unidades, con lo que hemos obtenido un nuevo tour que no recorre de forma extra la arista requerida e . Si z no recorre ninguna arista con c -coste 0-2 (o ambos nodos incidentes con e son de tipo \mathcal{O} y z recorre dos aristas de este tipo), e tendrá un c -coste de $l - 2$ y $F(z) = 2(l - 1)$ también. Duplicamos el ciclo con un c -coste adicional de $2(l - 1)$ y eliminamos las dos copias de e y de una de sus aristas adyacentes en el ciclo, con lo que disminuye el c -coste en $2(l - 2) + 2 = 2(l - 1)$, obteniendo así otro tour del mismo c -coste que no recorre e . Si tenemos un ciclo que recorre dos aristas requeridas de forma no consecutiva, podemos proceder de forma similar para obtener otro tour de igual o menor c -coste que no recorra (de forma extra) ninguna de las dos.

Como T es un árbol y todo lo que recorra x de modo extra ha de formar ciclos, la única posibilidad es que cada arista de T recorrida por x lo sea una vez en cada sentido. Tanto para las aristas con c -coste 1-1 como 0-2, el c -coste de recorrerlas una vez en cada sentido es 2. Como x debe visitar los $K - L$ vértices B_{L+1}, \dots, B_K y conectarlos con el conjunto $B_1 \cup B_2 \cup \dots \cup B_L$, ha de utilizar, al menos, $K - L$ aristas, lo que supondrá un c -coste de $2(K - L)$, que, añadido al de las aristas requeridas, suma un total de c_0 , con lo que se demuestra que la desigualdad Honeycomb₀₂ es válida.

◆

Teorema 5.6.2 *Las desigualdades Honeycomb₀₂ definen faceta de WGRP(G).*

Dem: Como en casos anteriores, demostraremos primero que las desigualdades Honeycomb₀₂ definen faceta de WGRP(G_C). Este poliedro tiene dimensión $2|E^C| - K + 1$. Sea $V_L = \{B_1, B_2, \dots, B_L\}$ y $V_K = \{B_{L+1}, B_{L+2}, \dots, B_K\}$. Comenzamos construyendo un ciclo que visite los nodos de V_L , pasando exactamente una vez por cada arista requerida y ninguna por las no requeridas.

Figura 5.25: Vector x^d (Honeycomb₀₂)

Como el grafo inducido por las aristas requeridas es conexo, y el grado de cada uno de los nodos de este grafo es par, podemos asegurar que tal ciclo existe. Llamaremos x^d al vector que representa a este ciclo (figura 5.25). Sea T el árbol que cumple las condiciones mencionadas en la descripción de la configuración Honeycomb₀₂. Denotaremos por T_1 al conjunto de aristas resultante de eliminar de T las aristas incidentes con los nodos de V_L , y $G(T_1)$ al subgrafo inducido por T_1 . A partir del vector x^d , podemos construir los mismos 5 tipos de tours cumpliendo la restricción con igualdad que en la demostración del teorema 5.5.1 para las desigualdades Honeycomb estándar, salvo por los de tipo 4. En lugar de éstos, dada una arista e en $E(V_L) \setminus T$ entre dos nodos B_i y B_j , construimos los siguientes tours:

- **Tipo 4a** Si B_i y B_j son ambos del mismo tipo (\mathcal{O} o \mathcal{I}), construimos un tour que recorra e de B_i a B_j y el camino entre B_j y B_i en T de B_j a B_i . El resto de aristas de T_1 las recorreremos en ambos sentidos. Además, podemos construir otro tour invirtiendo el sentido de recorrido de las aristas de éste. Estos tours son como los de tipo 4 para las desigualdades Honeycomb (figura 5.22).
- **Tipo 4b** Si B_i y B_j son de tipos distintos (supongamos que B_i es de tipo \mathcal{O} y B_j de tipo \mathcal{I}), y e está recorrida en x^d del nodo de tipo \mathcal{O} al de tipo \mathcal{I} (i.e., de B_i a B_j), construimos un tour (ver figura 5.26) haciendo 0 la componente de x^d que corresponde a recorrer e en dicho sentido y la recorreremos en sentido opuesto. Además, recorreremos dos veces las aristas del camino en T de B_i a B_j en este sentido. El resto de aristas en T_1 las recorreremos en ambos sentidos. Construimos un segundo tour en el que, además de lo recorrido por x^d , recorreremos la arista e de B_j a B_i , las aristas del camino en T de B_i a B_j una vez y el resto de aristas de T_1 una vez en cada sentido (ver figura 5.27).
- **Tipo 4c** Si B_i y B_j son de tipos distintos (B_i es de tipo \mathcal{O} y B_j de tipo

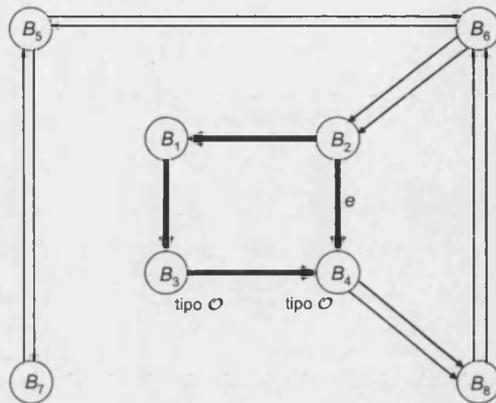


Figura 5.26: WGRP-tour 1 tipo 4b (Honeycomb₀₂)

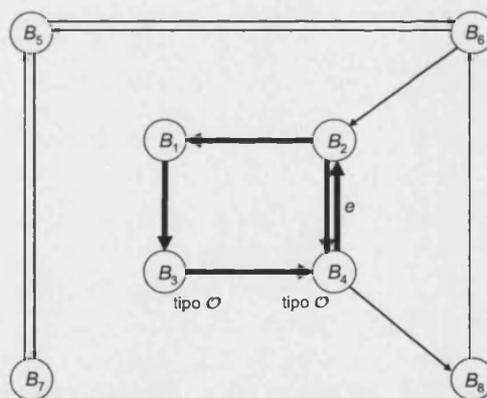
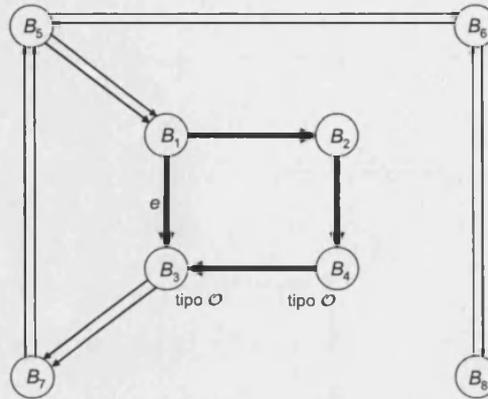
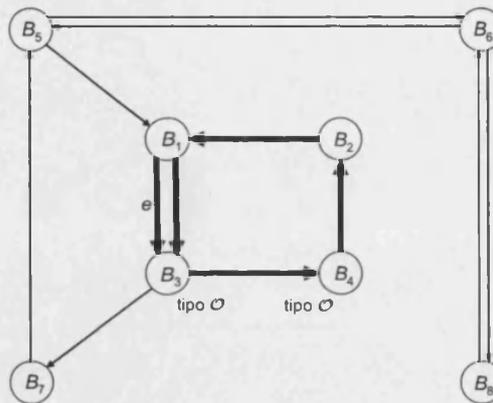


Figura 5.27: WGRP-tour 2 tipo 4b (Honeycomb₀₂)

Figura 5.28: WGRP-tour 3 tipo 4c (Honeycomb₀₂)Figura 5.29: WGRP-tour 4 tipo 4c (Honeycomb₀₂)

\mathcal{I}), y e está recorrida en x^d del nodo de tipo \mathcal{I} al de tipo \mathcal{O} (i.e., de B_j a B_i), construimos un tour (ver figura 5.28) invirtiendo el sentido de todas las aristas recorridas por x^d excepto e . También recorreremos las aristas del camino en T de B_i a B_j dos veces en dicho sentido, y las restantes de T_1 una vez en cada sentido. Construimos también un segundo tour recorriendo una segunda vez la arista e de B_j a B_i (además de la recorrida por x^d) y una vez las aristas del camino en T de B_i a B_j en ese sentido. Recorreremos también las aristas restantes de T_1 en ambos sentidos (ver figura 5.29).

- Tipo 4d** Si e es no requerida y B_i y B_j son de tipos distintos (supongamos que B_i es de tipo \mathcal{O} y B_j de tipo \mathcal{I}), construimos un tour (ver figura 5.30) que recorra e de B_i a B_j . Invertimos el sentido de recorrido de las aristas del camino de B_i a B_j recorrido por x^d y recorreremos una vez las aristas del camino de B_i a B_j en T en ese sentido. El resto de aristas de T_1 las

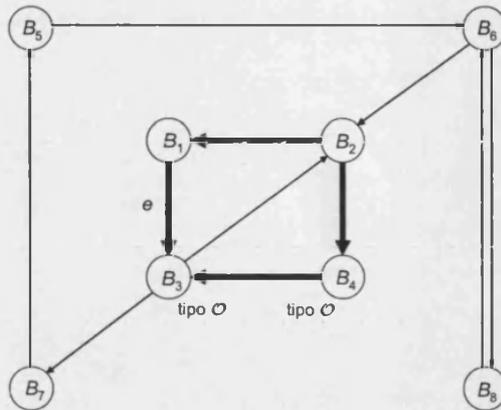


Figura 5.30: WGRP-tour 5 tipo 4d (Honeycomb₀₂)

recorremos en ambos sentidos. Además, construimos un tour que recorra e de B_j a B_i y el camino de B_i a B_j en T una vez en ese sentido, y el resto de aristas de T_1 en ambos sentidos. Este último tour es similar a los del tipo 4 para las desigualdades Honeycomb.

Si restamos el vector x^d a estos tours y a aquéllos descritos para las desigualdades Honeycomb estándar (excepto los de tipo 4) y los representamos en forma de matriz, obtenemos una matriz muy parecida a la de la demostración del teorema 5.5.1:

| | $T \setminus T_1$ | T_1 | $E(V_K) \setminus T$ | $(V_L \setminus V_K) \setminus T$ | $E_R(V_L)$ | $E_{NR}(V_L)$ |
|---------|-------------------|-------|----------------------|-----------------------------------|------------|---------------|
| $t5$ | * | A | 0 | 0 | 0 | 0 |
| $t1$ | B | 1 | 0 | 0 | 0 | 0 |
| $t2$ | * | * | I | 0 | 0 | 0 |
| $t3$ | * | * | 0 | I | 0 | 0 |
| $t4abc$ | * | * | 0 | 0 | C | 0 |
| $t4d$ | * | * | 0 | 0 | * | I |

Las filas $t1$, $t2$, $t3$ y $t5$ corresponden a los vectores de tipo 1,2,3 y 5, respectivamente, para las desigualdades Honeycomb estándar, mientras que las filas $t4abc$ y $t4d$ hacen referencia a los tours de tipo 4a, 4b y 4c, y 4d, respectivamente, contruidos específicamente para las desigualdades Honeycomb₀₂.

Puesto que ya vimos que la submatriz

$$\begin{pmatrix} * & A \\ B & 1 \end{pmatrix}$$

es de rango completo, sólo tenemos que comprobar que la submatriz C es de rango completo para demostrar que toda la matriz lo es. Llamaremos E_1 al



$$\left(\begin{array}{cccc|cccc} 0 & 1 & \dots & 1 & 1 & 0 & -1 & \dots & -1 & -1 \\ 1 & 0 & \dots & 1 & 1 & -1 & 0 & \dots & -1 & -1 \\ \vdots & & \ddots & & \vdots & \vdots & & \ddots & & \vdots \\ 1 & 1 & \dots & 0 & 1 & -1 & -1 & \dots & 0 & -1 \\ 1 & 1 & \dots & 1 & 0 & -1 & -1 & \dots & -1 & 0 \\ \hline & & & 0 & & & & & I & \end{array} \right)$$

Por lo tanto, las desigualdades Honeycomb₀₂ definen faceta de $\text{WGRP}(G_c)$ y, aplicando el teorema (3.4.3), de $\text{WGRP}(G)$.



Capítulo 6

Descripción del poliedro: Facetas Zigzag

Las facetas vistas hasta ahora no son suficientes por sí solas para describir completamente el poliedro de soluciones del WGRP. Existen soluciones fraccionarias que cumplen todas estas restricciones. Un ejemplo de esas soluciones se puede observar a continuación. La figura 6.1 representa el grafo y los costes de las aristas (la cifra representa el coste de recorrer la arista desde el nodo más cercano al nodo opuesto, y las aristas requeridas están representadas en negrita). Nótese que los nodos de esta instancia bien pudieran ser conjuntos de nodos de un grafo mayor, por eso tiene sentido que aparezcan aristas requeridas paralelas. En la figura 6.2 podemos ver la solución óptima x^* (la líneas discontinuas representan a aristas recorridas con 0.5 en la solución) de la relajación lineal del problema WGRP asociado, que tiene coste 10. Esta solución fraccionaria cumple todos los tipos de restricciones descritos hasta el momento. Sin embargo, la solución óptima (Figura 6.3) tiene coste 11. Es fácil ver que la solución fraccionaria x^* incumple la siguiente restricción:

$$F(x) = x_{14} + x_{41} + x_{13} + x_{31} + x'_{13} + x'_{31} + 2x_{21} + 2x_{43} + \\ + x_{23} + x_{32} + x_{24} + x_{42} + x'_{24} + x'_{42} \geq 6$$

pues $F(x^*) = 5$. Otro ejemplo de instancia con una solución fraccionaria puede verse en las figuras 6.4 y 6.5. En este caso, las aristas requeridas recorridas con 1.5 están representadas por dos arcos, uno continuo y otro discontinuo, mientras que las aristas discontinuas con flecha en ambos extremos representan a aristas con $x_{ij}^* = x_{ji}^* = 0,5$. La solución fraccionaria tiene un coste total de 38, mientras que el coste de una solución óptima es 42 (figura 6.6). La desigualdad violada en este caso sería:

$$F(x) = x_{14} + x_{41} + x_{13} + x_{31} + 2x_{21} + 2x_{43} + x_{23} + x_{32} + x_{24} + x_{42} \geq 6$$

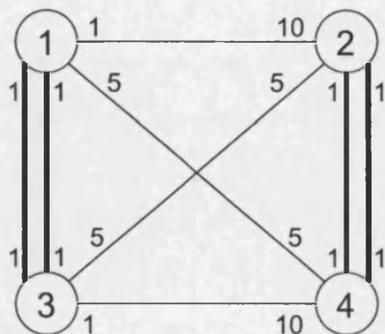


Figura 6.1: Instancia del WGRP con solución fraccionaria 1

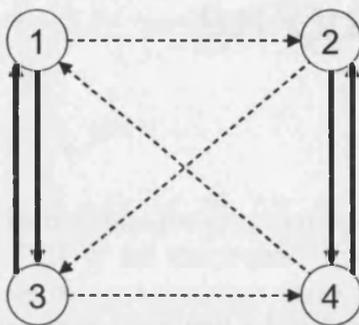


Figura 6.2: Solución fraccionaria 1

, pues $F(x^*) = 5$.

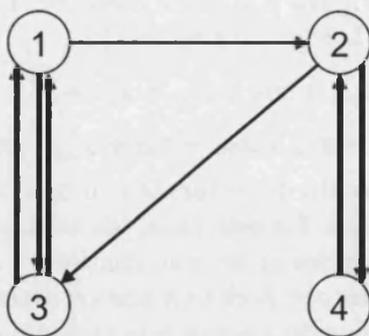


Figura 6.3: Solución óptima 1

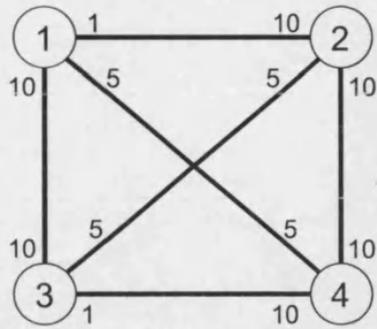


Figura 6.4: Instancia del WGRP con solución fraccionaria 2

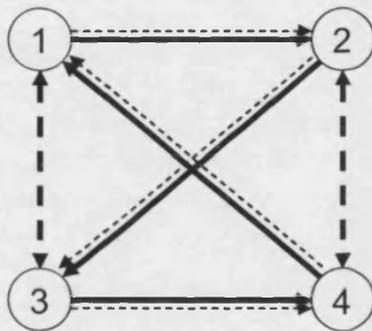


Figura 6.5: Solución fraccionaria 2

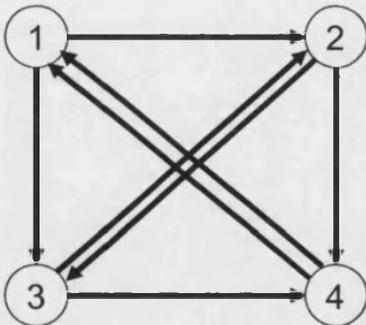


Figura 6.6: Solución óptima 2

Estas restricciones pertenecen a una nueva familia de desigualdades, a las que llamaremos desigualdades Zigzag. Este nombre viene de la forma de las soluciones fraccionarias que son cortadas por las desigualdades. Estas soluciones tienen la característica de tener un *zigzag* de aristas recorridas con 0.5 *extra*, i.e., $x_{ij} + x_{ji} = 0,5$ si (i, j) es no requerida, y $x_{ij} + x_{ji} = 1,5$ si (i, j) es requerida (figuras 6.2 y 6.5). Dentro de la familia de desigualdades Zigzag distinguiremos dos tipos, las desigualdades Zigzag pares, cuando el número de aristas requeridas incidentes con cada nodo es par (figura 6.1), y Zigzag impares, cuando el número de aristas requeridas incidentes con cada nodo es impar (figura 6.4). A continuación veremos cuál es la expresión de cada uno de estos tipos de desigualdad. Veremos, también, que son válidas para el WGRP (aunque por razones distintas en cada caso) y que inducen faceta de WGRP(G).

6.1. Desigualdades Zigzag pares

Sea $G = (V, E)$ un grafo. Consideremos una partición de V en 4 conjuntos M^1, M^2, M^3 y M^4 de manera que los subgrafos $G(M^i)$ sean conexos y $\delta_R(M^i)$ sea impar, es decir, el número de aristas requeridas incidentes con un nodo en M^i y otro fuera de M^i es impar (diremos entonces que M^i es *R-impar*). Sea $\alpha_{ij} = |(M^i : M^j)_R|$ el número de aristas requeridas entre un par de subconjuntos M^i y M^j . Supongamos que se cumple

$$\alpha_{12} = \alpha_{34} = \alpha_{14} = \alpha_{23} = 0 \quad (6.1)$$

es decir, no hay aristas requeridas en la cortadura $(M^1 \cup M^3 : M^2 \cup M^4)$.

La desigualdad Zigzag par se define entonces mediante los siguientes coeficientes (ver figura 6.7):

- $c(M^1, M^2) = c(M^3, M^4) = 0$.
- $c(M^2, M^1) = c(M^4, M^3) = 2$.
- $c(M^i, M^j) = 1$ en otro caso.

Dados dos conjuntos de nodos S_1 y S_2 , y el conjunto de aristas entre ellos $(S_1 : S_2)$, denotamos $x(S_1 : S_2) = \sum_{(i,j): i \in S_1, j \in S_2} x_{ij}$. De este modo, llamaremos *desigualdad zigzag par* a la siguiente desigualdad:

$$F(x) = x(\delta(M^1 \cup M^2)) + 2x(M^2 : M^1) + 2x(M^4 : M^3) \geq \alpha_{13} + \alpha_{24} + 2 \quad (6.2)$$

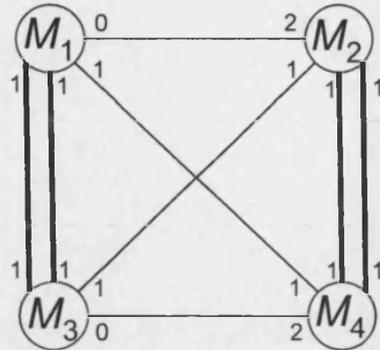


Figura 6.7: Grafo de configuración asociado a una desigualdad Zigzag par

Teorema 6.1.1 *La desigualdad Zigzag par (6.2) es válida para $WGRP(G)$.*

Dem: Basta demostrar que la desigualdad es válida para $WGRP(G_C)$. Sea x un tour del WGRP en G_C . Como hay $\alpha_{13} + \alpha_{24}$ aristas requeridas en G_C , todas ellas con coeficiente 1 en la desigualdad, tenemos que $F(x) = \alpha_{13} + \alpha_{24} + \alpha_{23} + \alpha_{14} + \Delta$, con $\Delta \geq 0$. Al no haber aristas requeridas en la cortadura $\delta(M^1 \cup M^3, M^2 \cup M^4)$, x habrá de cruzar dicha cortadura, al menos una vez en cada sentido, utilizando aristas no requeridas, ya que la solución debe ser conexa. La única manera de hacer esto con c -coste menor que 2 es utilizando una arista con coeficiente 0 y otra con coeficiente 1, pero esto haría que dos nodos tuviesen grado impar en la solución, con lo que habría que recorrer una tercera arista con coeficiente 1, y así $\Delta \geq 2$. ♦

Teorema 6.1.2 *Si G_C es completo y $\alpha_{13}, \alpha_{24} \geq 2$, la desigualdad Zigzag par (6.2) define faceta de $WGRP(G)$.*

Dem: Veamos en primer lugar que la desigualdad Zigzag par define faceta de $WGRP(G_C)$. La dimensión de $WGRP(G_C)$ es $\dim(WGRP(G_C)) = 2(\alpha_{13} + \alpha_{24}) + 5$, por lo que este será el número de tours linealmente independientes que deberemos construir.

Como α_{13} y α_{24} son pares, podemos orientar las aristas requeridas de manera que formen un subgrafo dirigido simétrico (figura 6.8). Llamaremos x^d al vector que representa a dicho grafo. Primero, seleccionamos las 4 componentes asociadas a las aristas del zigzag e_{12} , e_{23} , e_{34} y e_{14} en el sentido definido por el zigzag $M_1 - M_2 - M_3 - M_4 - M_1$ (figura 6.9). Para cada componente no seleccionada, construiremos un tour que utilice únicamente dicha componente más las componentes seleccionadas y el vector x^d . Además, construiremos un tour más empleando solamente el vector x^d y componentes seleccionadas. Dada una arista $e_{ij} = (i, j)$, llamaremos $y(e_{ij}^{\rightarrow})$ ($y(e_{ij}^{\leftarrow})$) al vector que tiene un 1 en la componente correspondiente a la arista e_{ij} en sentido de i a j (de j a i) y 0 en

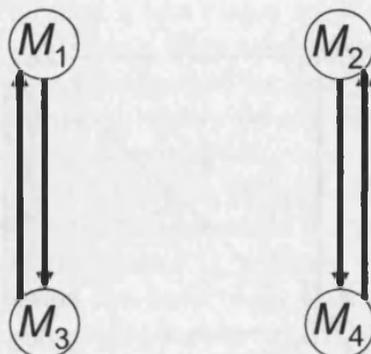
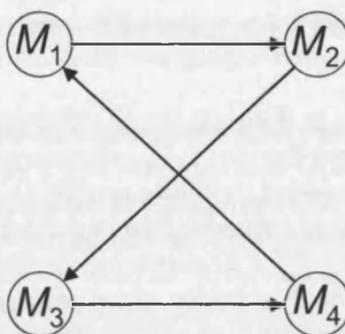
Figura 6.8: Vector x^d , caso par

Figura 6.9: Componentes seleccionadas, caso par

el resto de componentes. Partiendo del vector x^d , lo completamos para obtener dichos tours de la siguiente forma:

- Para cada arista $e \in \delta(M_1 \cup M_3)$, construimos el tour $x^d + y(\vec{e}_{ij}) + y(\vec{e}_{ij})$, que recorre dicha arista una vez en cada sentido (figura 6.10).
- Para cada arista $e_{13} \in (M_1 : M_3)$, construimos los tours $x^d + y(\vec{e}_{13}) + y(\vec{e}_{34}) + y(\vec{e}_{14})$ y $x^d + y(\vec{e}_{13}) + y(\vec{e}_{12}) + y(\vec{e}_{23})$ (figuras 6.11 y 6.12).
- Construimos el tour $x^d + y(\vec{e}_{12}) + y(\vec{e}_{23}) + y(\vec{e}_{34}) + y(\vec{e}_{14})$, que recorre una vez el zigzag (figura 6.13).

Si restamos el vector x^d de todos estos tours, que cumplen $F(x) = c_0$, y los escribimos en una matriz, obtenemos:

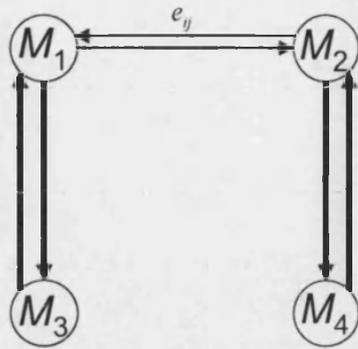


Figura 6.10: WGRP-tour 1 (Zigzag par)

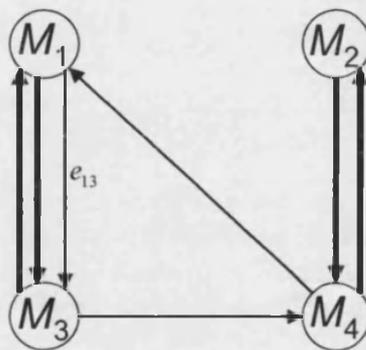


Figura 6.11: WGRP-tour 2 (Zigzag par)

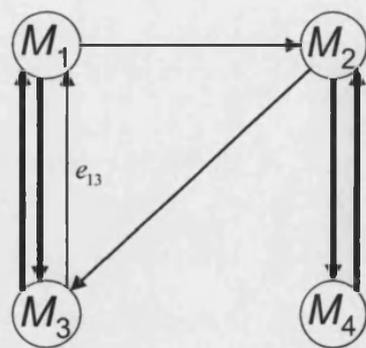


Figura 6.12: WGRP-tour 3 (Zigzag par)

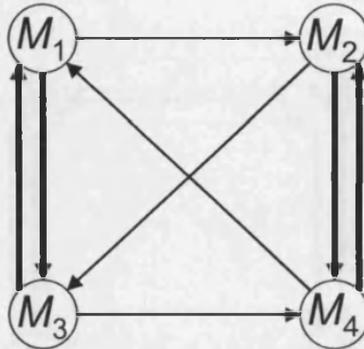


Figura 6.13: WGRP-tour 4 (Zigzag par)

$$\begin{array}{c}
 \text{componentes} \\
 \text{no seleccionadas}
 \end{array}
 \begin{array}{cccc}
 \xrightarrow{x_{12}} & \xrightarrow{x_{23}} & \xrightarrow{x_{34}} & \xleftarrow{x_{14}}
 \end{array}$$

$$\left(\begin{array}{ccc|cccc}
 & \text{I} & & & & & * & & & \\
 \hline
 & 0 & & & & 1 & 1 & 1 & 1 &
 \end{array} \right)$$

Esta matriz es de rango completo, por lo que los vectores correspondientes a los tours que hemos construido son afinmente independientes. Con ello queda probado que la desigualdad Zigzag induce una faceta de $\text{WGRP}(G_C)$. Aplicando el teorema 3.4.3, llegamos a que también induce una faceta de $\text{WGRP}(G)$. \blacklozenge

6.2. Desigualdades Zigzag impares

En este caso, consideraremos una partición del conjunto de nodos V similar a la del caso anterior, pero ahora los cuatro subconjuntos contendrán un número impar de vértices R -impares, es decir, $\delta_R(M_i)$ es impar (figura 6.14). Definimos los siguientes conjuntos de aristas:

- $\mathcal{H} = (M^1, M^2) \cup (M^3, M^4)$, es decir, las aristas *horizontales* en la figura 6.15.
- $\mathcal{D} = (M^2, M^3) \cup (M^1, M^4)$, es decir, las aristas *diagonales* en la figura 6.15.

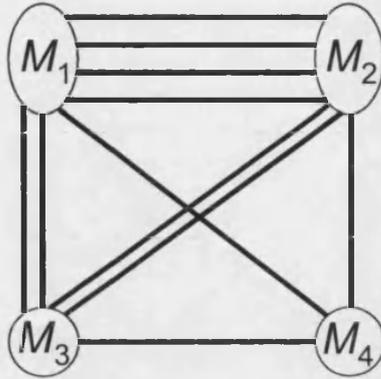
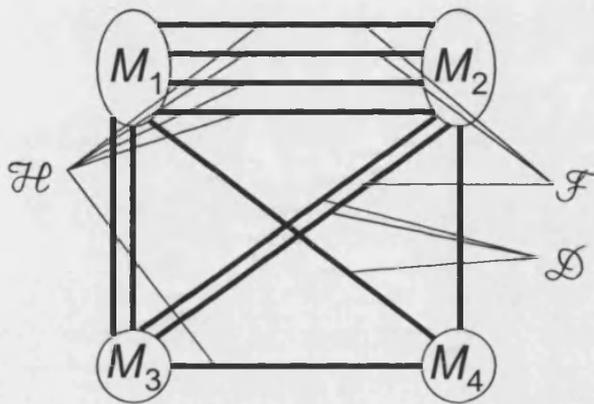


Figura 6.14: Partición del grafo, desigualdades Zigzag impares

Figura 6.15: Conjuntos \mathcal{H} , \mathcal{D} y \mathcal{F}

Nótese que la unión de estos dos conjuntos es la cortadura entre los conjuntos $M^1 \cup M^3$ y $M^2 \cup M^4$, i.e., $\mathcal{H} \cup \mathcal{D} = \delta(M^1 \cup M^3)$. Sea ahora \mathcal{F} un conjunto cualquiera de aristas requeridas contenido estrictamente en $\mathcal{H} \cup \mathcal{D}$, es decir, $\mathcal{F} \subset (\mathcal{H} \cup \mathcal{D})_R$ (ver figura 6.15), cumpliendo:

$$|\mathcal{H}_R \setminus \mathcal{F}| + |\mathcal{D}_R \cap \mathcal{F}| = |\mathcal{D}_R \setminus \mathcal{F}| + |\mathcal{H}_R \cap \mathcal{F}| \quad (6.3)$$

Para entender esta condición, supongamos que orientamos las aristas horizontales de \mathcal{F} de derecha a izquierda, es decir, de $M^2 \cup M^4$ a $M^1 \cup M^3$, y las diagonales de \mathcal{F} al revés, de izquierda a derecha. Las aristas que no están en \mathcal{F} las consideraremos orientadas en sentido opuesto (las horizontales de izquierda a derecha y las diagonales de derecha a izquierda). Entonces, la condición (6.3) quiere decir que, con esta orientación, la cortadura $\delta_R(M^1 \cup M^3)$ está equilibrada. Obsérvese que las aristas de \mathcal{F} tendrían una orientación opuesta al zigzag que se recorre con 0.5 extra en las soluciones fraccionarias comentadas anteriormente.

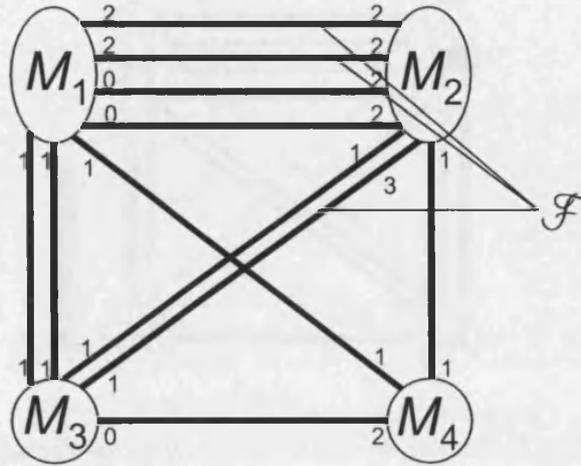


Figura 6.16: Grafo de configuración asociado a una desigualdad Zigzag impar

El grafo de configuración G_C asociado a las desigualdades Zigzag impares está definido por la partición descrita, el conjunto de aristas \mathcal{F} y los siguientes coeficientes asociados a cada arista e_{ij} (figura 6.16):

$$(c_{ij}, c_{ji}) = \begin{cases} (0, 2) & \forall e_{ij} \in \mathcal{H} \setminus \mathcal{F}, i \in M^1 \cup M^3, j \in M^2 \cup M^4 \\ (2, 2) & \forall e_{ij} \in \mathcal{H} \cap \mathcal{F} \\ (1, 3) & \forall e_{ij} \in \mathcal{D} \cap \mathcal{F}, i \in M^1 \cup M^3, j \in M^2 \cup M^4 \\ (1, 1) & \text{en otro caso} \end{cases}$$

Con estos coeficientes, la desigualdad Zigzag impar es:

$$\begin{aligned} & x(\delta(M^1 \cup M^2)) + 2x(M^2 : M^1) + 2x(M^4 : M^3) + \\ & + 2x(\mathcal{F}_{zz}) \geq \alpha_{13} + \alpha_{24} + \alpha_{14} + \alpha_{23} + 2|\mathcal{H} \cap \mathcal{F}| + 2 \end{aligned} \quad (6.4)$$

donde $x(\mathcal{F}_{zz})$ representa la suma de las componentes asociadas a las aristas en \mathcal{F} en la dirección del zigzag.

Como puede verse, esta desigualdad no cumple todas las condiciones para ser considerada una desigualdad de configuración, ya que en el grafo de configuración puede haber aristas paralelas con coeficientes distintos en la desigualdad. Son, por tanto, desigualdades de configuración débiles.

Sin embargo, bajo determinadas circunstancias, sí pueden ser desigualdades de configuración. En concreto, cuando $\mathcal{F} = \emptyset$ (lo cual implica, por (6.3), que $|\mathcal{H}_R| = |\mathcal{D}_R|$) sí lo es. En este caso particular, los coeficientes de la desigualdad Zigzag impar correspondiente son iguales a los de las desigualdades Zigzag pares.

Teorema 6.2.1 *La desigualdad Zigzag impares (6.4) es válida para WGRP(G).*

Dem: Sea $F(x) \geq c_0$ la desigualdad Zigzag impar. Basta demostrar que la desigualdad es válida para WGRP(G_C). Sea x un tour del WGRP en G_C . Puesto que las $\alpha_{13} + \alpha_{24} + \alpha_{14} + \alpha_{23}$ aristas requeridas en $\delta(M^1 \cup M^2)$ tienen coeficiente mayor o igual que 1, y que todas las aristas de $\mathcal{H} \cap \mathcal{F}$ tienen coeficiente 2, se cumple $F(x^*) \geq c_0 - 2$. Si x^* recorre alguna de las arista requeridas de $\mathcal{H} \setminus \mathcal{F}$ o $\mathcal{D} \cap \mathcal{F}$ de derecha a izquierda, es decir, de $M_2 \cup M^4$ a $M_1 \cup M^3$, sumará, al menos, dos unidades más, con lo que se cumplirá la desigualdad. Supongamos pues que todas estas aristas están recorridas exactamente una vez de izquierda a derecha. Entonces, por la condición (6.3), x^* recorrerá todas las aristas requeridas en $\mathcal{D}_R \setminus \mathcal{F}$ y $\mathcal{H}_R \cap \mathcal{F}$ de derecha a izquierda. Como los cuatro vértices de G_C son R-impares, necesitamos añadir un matching. El único matching con \lfloor -coste menor que 2 es el formado por una arista entre M^1 y M^2 y otra entre M^3 y M^4 no incluidas en \mathcal{F} , recorridas ambas de izquierda a derecha. Pero en este caso, tenemos que recorrer dos aristas más de derecha a izquierda para equilibrar la cortadura $\delta(M^1 \cup M^3)$, con lo que tenemos las dos unidades extras y la desigualdad también se cumple. \blacklozenge

Teorema 6.2.2 *La desigualdad Zigzag $F(x) \geq c_0$ (6.4) define una faceta de WGRP(G) si $G_C \setminus \mathcal{F}$ es un grafo completo, existen dos aristas requeridas $e_{13} \in (M^1, M^3)$ y $e_{24} \in (M^3, M^4)$ y las aristas requeridas restantes pueden orientarse de manera que induzcan un grafo simétrico que satisfaga:*

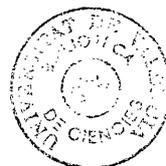
- *todas las aristas en $(\mathcal{H} \cup \mathcal{D})_R \setminus \mathcal{F}$ están orientadas en el sentido del zigzag $M^1-M^2-M^3-M^4-M^1$.*
- *todas las aristas de \mathcal{F} están orientadas en el sentido opuesto a dicho zigzag.*

Dem: Veamos que la desigualdad induce una faceta de WGRP(G_C). Para ello, necesitamos construir $2|E^C| - 3$ tours linealmente independientes que la satisfagan con igualdad.

Sea x^d el vector de incidencia que representa al grafo simétrico mencionado en el enunciado del teorema (figura 6.17). Es fácil comprobar que dicho vector tiene un c -coste de $c_0 - 2$.

Seleccionaremos en primer lugar las cuatro componentes asociadas a las aristas e_{13} y e_{24} . Obsérvese que estas cuatro componentes valen 0 en el vector x^d . Como $G_C \setminus \mathcal{F}$ es completo, podemos seleccionar otras cuatro componentes asociadas a 4 aristas $e_{12} \in (M^1, M^2)$, $e_{23} \in (M^2, M^3)$, $e_{34} \in (M^3, M^4)$ y $e_{14} \in (M^1, M^4)$ no incluidas en \mathcal{F} . En particular, para cada una de estas aristas seleccionamos la componente asociada a la dirección dada por el zigzag (figura 6.18).

Dada una arista $e \in (M^i, M^j)$, denotaremos $y(\vec{e})$ ($y(\overleftarrow{e})$) al vector con 1 en la componente asociada a la arista e en sentido de i a j (de j a i) y 0 en el resto. Para cada componente no seleccionada, vamos a construir un tour que



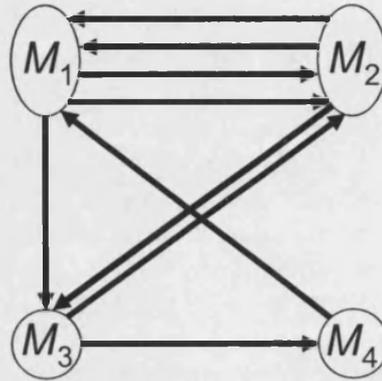
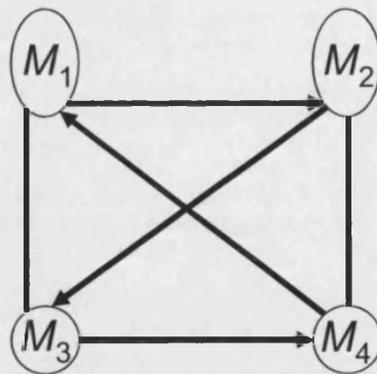
Figura 6.17: Vector x^d , caso impar

Figura 6.18: Componentes seleccionadas, caso impar

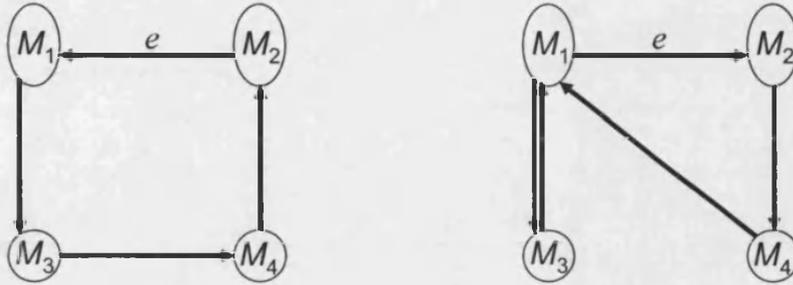


Figura 6.19: WGRP-tours 1 (Zigzag impar)

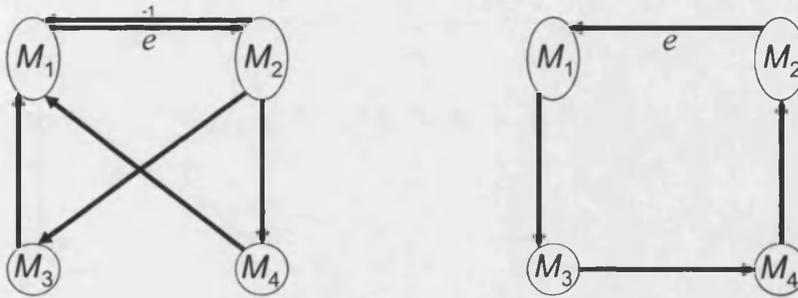


Figura 6.20: WGRP-tours 2 (Zigzag impar)

utiliza dicha componente exactamente una vez, las utilizadas por el vector x^d y componentes seleccionadas:

- Para cada arista $e \in (M^1, M^2) \setminus \mathcal{F}$, construimos el tour $x = x^d + y(\vec{e}) + y(\vec{e}_{13}) + y(\vec{e}_{34}) + y(\vec{e}_{24})$ y, si la componente de M^1 a M^2 no está seleccionada, el tour $x = x^d + y(\vec{e}) + y(\vec{e}_{24}) + y(\vec{e}_{14}) + y(\vec{e}_{13}) + y(\vec{e}_{13})$ (figura 6.19).
- Para cada arista $e \in (M^1, M^2) \cap \mathcal{F}$, construimos el tour $x = x^d + y(\vec{e}) - y(\vec{e}) + y(\vec{e}_{24}) + y(\vec{e}_{14}) + y(\vec{e}_{23}) + y(\vec{e}_{13})$ y el tour $x = x^d + y(\vec{e}) + y(\vec{e}_{13}) + y(\vec{e}_{34}) + y(\vec{e}_{24})$ (figura 6.20).
- Para cada arista $e \in (M^1, M^3)$ cuyas componentes no hayan sido seleccionadas, construimos los tours $x = x^d + y(\vec{e}) + y(\vec{e}_{13}) + y(\vec{e}_{12} + y(\vec{e}_{24})) + y(\vec{e}_{14})$ y $x = x^d + y(\vec{e}) + y(\vec{e}_{13}) + y(\vec{e}_{12}) + y(\vec{e}_{24}) + y(\vec{e}_{14})$ (figura 6.21).
- Para cada arista $e \in (M^2, M^3) \setminus \mathcal{F}$, construimos el tour $x = x^d + y(\vec{e}) + y(\vec{e}_{24}) + y(\vec{e}_{14}) + y(\vec{e}_{13})$ y, si la componente de M^2 a M^3 no está seleccionada, el tour $x = x^d + y(\vec{e}) + y(\vec{e}_{34}) + y(\vec{e}_{24}) + y(\vec{e}_{13}) + y(\vec{e}_{13})$ (figura 6.22).
- Para cada arista $e \in (M^2, M^3) \cap \mathcal{F}$, construimos el tour $x = x^d + y(\vec{e}) - y(\vec{e}) + y(\vec{e}_{34}) + y(\vec{e}_{24}) + y(\vec{e}_{13}) + y(\vec{e}_{12})$ y el tour $x = x^d + y(\vec{e}) + y(\vec{e}_{24})$

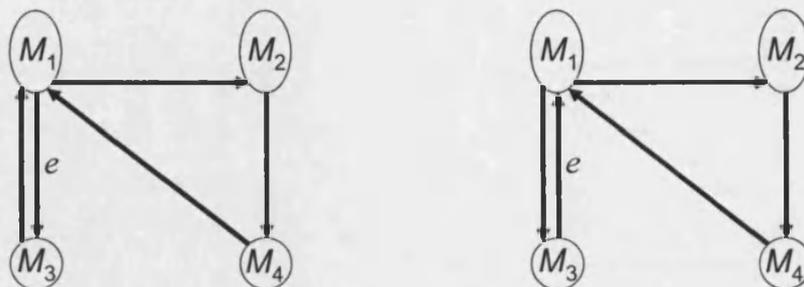


Figura 6.21: WGRP-tours 3 (Zigzag impar)

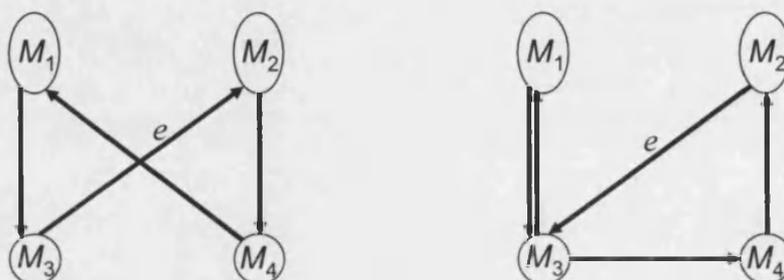


Figura 6.22: WGRP-tours 4 (Zigzag impar)

) + $y(e_{14}) + y(e_{13})$ (figura 6.23).

Para las componentes no seleccionadas de las aristas en (M^3, M^4) , (M^2, M^4) y (M^1, M^4) construiremos otros tantos tours de forma análoga. Por último, construiremos 5 tours más utilizando únicamente x^d y las componentes seleccionadas. Estos 5 tours pueden verse en la figura 6.24.

Todos estos tours cumplen $F(x) = c_0$. Si les restamos el vector x^d y los expresamos en forma de matriz, obtenemos:

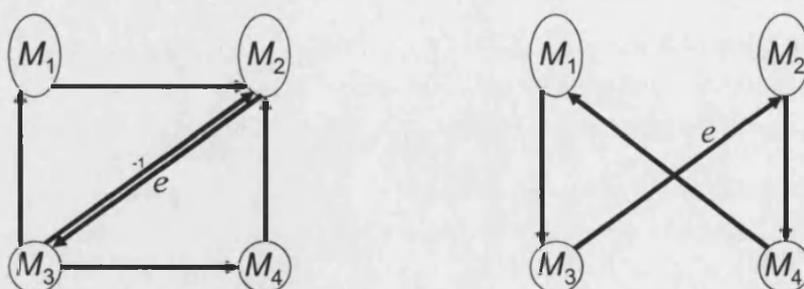


Figura 6.23: WGRP-tours 5 (Zigzag impar)

6.3. Relación de las nuevas facetas con otros problemas similares 109

| \mathcal{F} | Componentes no | Componentes seleccionadas |
|---|---------------------------------------|---|
| | seleccionadas $\setminus \mathcal{F}$ | (1,3) (3,1) (2,4) (4,2) (1,2) (2,3) (3,4) (4,1) |
| $\begin{pmatrix} 1 & -1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & -1 & \dots & 0 & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 & 0 \\ & & & & \ddots & & \\ 0 & 0 & 0 & 0 & \dots & 1 & -1 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix}$ | 0 | * |
| 0 | I | * |
| 0 | 0 | $\begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 2 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$ |

Es fácil ver que la submatriz inferior derecha es de rango completo, por lo que también lo es la matriz de todos los vectores. De este modo, los tours son afinmente independientes, con lo que queda demostrado que la desigualdad define una faceta de $WGRP(G_C)$. Aplicando el teorema 3.4.4, también define una faceta de $WGRP(G)$. ♦

6.3. Relación de las nuevas facetas con otros problemas similares

Bajo ciertas condiciones, la familia de desigualdades Zigzag puede extenderse a otros problemas de Rutas por Arcos en grafos Dirigidos, No Dirigidos y Mixtos.

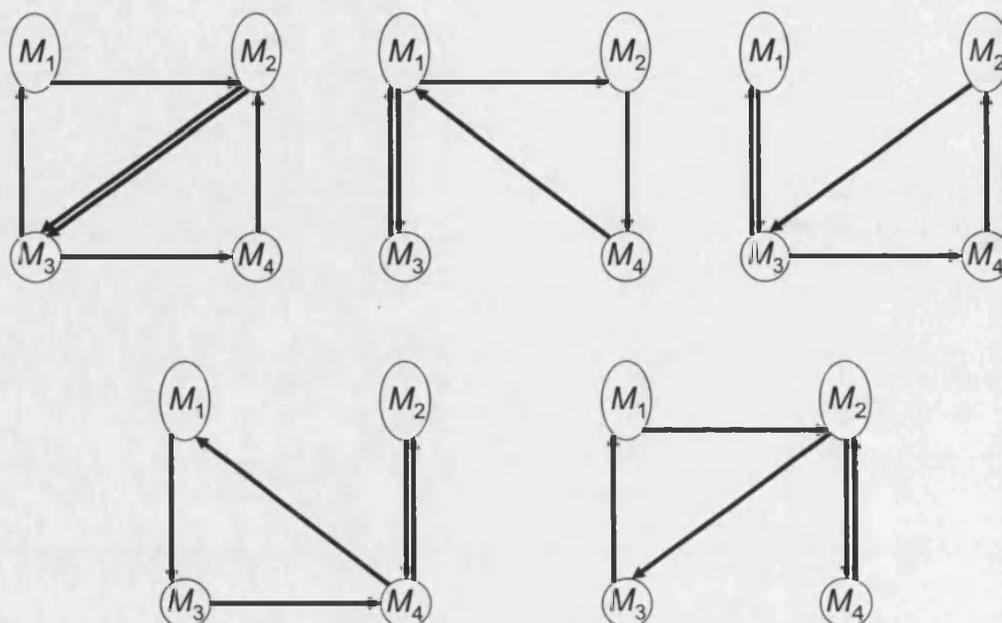


Figura 6.24: WGRP-tours 6 (Zigzag impar)

6.3.1. Las desigualdades Zigzag en el WPP

Cuando todas las aristas del grafo G son requeridas, el WGRP se reduce al WPP. Es obvio que en este caso las desigualdades Zigzag pares no existen, ya que es necesario que las aristas de la cortadura $\delta(M^1 \cup M^3)$ sean no requeridas. Las desigualdades Zigzag impares, sin embargo, son válidas y definen faceta de $WPP(G)$ bajo las mismas condiciones que en el WGRP. Cabe destacar que, cuando $\mathcal{F} = \emptyset$, las desigualdades Zigzag impares son equivalentes a las desigualdades *3-wheel* introducidas por Win (1987). En la figura 6.25 podemos ver un ejemplo de cómo convertir una desigualdad Zigzag impar en una *3-wheel*. La desigualdad Zigzag impar sería:

$$2x_{21} + 2x_{43} + x_{13} + x_{31} + x_{14} + x_{41} + x_{23} + x_{32} + x_{24} + x_{42} \geq 6$$

Utilizando las ecuaciones de grado sobre los nodos M^2 y M^1 , podemos transformarla en:

$$2x_{12} + 2x_{13} + 2x_{14} + 2x_{32} + 2x_{42} + 2x_{43} \geq 6$$

Dividimos la desigualdad por 2 y obtenemos una desigualdad *3-wheel*:

$$x_{12} + x_{13} + x_{14} + x_{32} + x_{42} + x_{43} \geq 3$$

Nótese, sin embargo, que en el caso en que $\mathcal{F} \neq \emptyset$, las desigualdades Zigzag impares son nuevas para el WPP.

6.3. Relación de las nuevas facetas con otros problemas similares 111

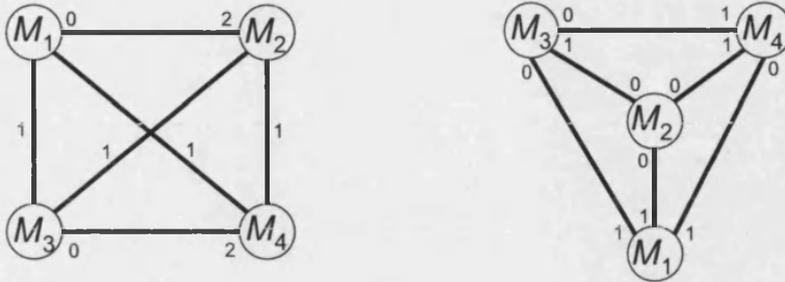


Figura 6.25: 3-Wheel en forma de Zigzag

6.3.2. Las desigualdades Zigzag en grafos no dirigidos

En el caso del Problema General de Rutas (GRP), si trabajamos con una formulación que emplee dos variables para cada arista, las desigualdades Zigzag pueden aplicarse directamente. De manera análoga al caso del WGRP, se puede demostrar que estas desigualdades son válidas e inducen facetas del poliedro. Sin embargo, no son necesarias para resolver este tipo de instancias, ya que las soluciones fraccionarias cortadas por las Zigzag no aparecen en este caso. Dichas soluciones contienen un zigzag donde las variables que lo forman toman el valor 0.5. Pero, dado que en el GRP los costes de las aristas son iguales en ambos sentidos, estas soluciones no aparecen. Por otra parte, la formulación natural del GRP utiliza únicamente una variable.

6.3.3. Las desigualdades Zigzag en grafos dirigidos

Cualquier instancia del Problema General de Rutas en un Grafo Dirigido (DGRP) puede transformarse en una instancia del WGRP. Para ello, sólo tenemos que representar cada arco mediante una arista con coste infinito en la dirección opuesta a la del arco. Como cualquier solución del problema en un grafo dirigido también es solución de su versión para el WGRP, cumplirá las desigualdades Zigzag (las variables correspondientes a las aristas en la dirección opuesta al arco asociado desaparecen en la versión dirigida). Por tanto, las desigualdades Zigzag son válidas para los problemas en grafos dirigidos. Sin embargo, las desigualdades Zigzag impares no definen faceta del poliedro, ya que están dominadas por las desigualdades de flujo. Además, no es posible obtener soluciones fraccionarias como las que son cortadas por las desigualdades Zigzag impares. Recordemos que en estas soluciones hay dos aristas que son recorridas en ambos sentidos con valor 0.5. Como ahora no tenemos aristas, sino arcos, esto no es posible. Así pues, las únicas desigualdades Zigzag que tienen sentido en los

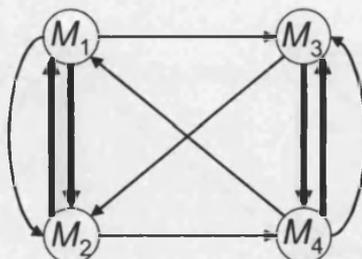


Figura 6.26: Grafo de configuración para las Zigzag en el DRPP

grafos dirigidos son las pares. Esto implica que ha de haber arcos no requeridos, con lo que las desigualdades Zigzag no tendrán sentido para el Problema del Cartero Chino en un Grafo Dirigido (DCPP). Veamos que, en el caso par, las desigualdades Zigzag definen facetas del $DGRP(G)$. En este caso, α_{ij} representa el número de arcos requeridos entre los conjuntos M^i y M^j (sin tener en cuenta el sentido).

Teorema 6.3.1 *Las desigualdades Zigzag pares $F(x) \geq c_0 = \alpha_{13} + \alpha_{24} + 2$ inducen faceta de $DGRP(G)$ si se satisfacen las condiciones:*

- *Existen, al menos, cuatro arcos $a_{12} \in (M_1 : M_2)$, $a_{23} \in (M_2 : M_3)$, $a_{34} \in (M_3 : M_4)$ y $a_{41} \in (M_4 : M_1)$ formando el zigzag.*
- *La mitad de los α_{13} arcos requeridos entre M_1 y M_3 van de M_1 a M_3 , y la otra mitad de M_3 a M_1 .*
- *La mitad de los α_{24} arcos requeridos entre M_2 y M_4 van de M_2 a M_4 , y la otra mitad de M_4 a M_2 .*

Dem: La dimensión de $DGRP(G)$ es $\dim(DGRP(G)) = |A| - |V| + 1$ (Corberán, Romero y Sanchis, 2003). Si para cada arco construimos el mismo tour que para la correspondiente componente de la arista asociada en el $WGRP(G)$, estos tours cumplen la desigualdad Zigzag con igualdad y siguen siendo linealmente independientes. \blacklozenge

6.3.4. Las desigualdades Zigzag en grafos mixtos

El Problema General de Rutas en un Grafo Mixto puede formularse de dos formas distintas. La primera formulación utiliza dos variables (Christofides et al. 1984, Ralphs 1993), y es análoga a la presentada para el $WGRP$. En este caso,

6.3. Relación de las nuevas facetas con otros problemas similares 113

las desigualdades Zigzag son muy similares a las del WGRP. La segunda formulación, propuesta por Nobert y Picard (1996) y Corberán et al. (2003,2005aa), emplea una variable para cada enlace.

En el MGRP, cada arista no requerida puede ser sustituida por dos arcos no requeridos en sentidos opuestos, así que podemos suponer que $E_R = E$.

A partir de ahora, nos referiremos a la formulación que utiliza una variable para cada arista como formulación F1, mientras que denotaremos F2 a la formulación con dos variables por arista. De la misma forma, llamaremos $MGRP_{F1}(G)$ y $MGRP_{F2}(G)$ a los poliedros asociados al conjunto de vectores solución del MGRP con cada una de las formulaciones. Corberán, Mota y Sanchis (2005bb) presentan una comparación de ambas formulaciones.

Si utilizamos la formulación F2, resulta obvio que las desigualdades Zigzag son válidas para $MGRP_{F2}(G)$. Como podemos ver en el siguiente teorema, también definen faceta de $MGRP_{F2}(G)$.

Teorema 6.3.2 *Las desigualdades Zigzag inducen faceta de $MGRP_{F2}(G)$ si existen cuatro enlaces definiendo el Zigzag $M^1-M^2-M^3-M^4-M^1$ y se cumple:*

- *en el caso par: las aristas requeridas en G_C pueden orientarse de manera que los enlaces requeridos induzcan un grafo simétrico.*
- *en el caso impar: existen dos aristas requeridas $e_{13} \in (M^1, M^3)$ y $e_{24} \in (M^2, M^4)$, y el resto de aristas requeridas en G_C pueden orientarse de manera que los enlaces requeridos induzcan un grafo simétrico en el que todos los enlaces en $(\mathcal{H} \cup \mathcal{D})_{\mathcal{R}} \setminus \mathcal{F}$ estén orientados en la dirección del Zigzag y todos los enlaces de \mathcal{F} estén orientados en dirección opuesta.*

Dem: La demostración es similar a la de los teoremas 6.1.2 y 6.2.2 para el WGRP.



Consideremos ahora la formulación F1, que sólo emplea una variable por cada arista. Para presentar esta formulación, dados dos conjuntos de vértices $S_1, S_2 \in V$, utilizaremos la siguiente notación:

- $A^+(S_1)$: conjunto de arcos que salen del conjunto de vértices S_1 .
- $A^-(S_2)$: conjunto de arcos que entran en el conjunto de vértices S_1 .
- $E(S_1)$: conjunto de aristas con un extremo en S_1 y otro en $V \setminus S_1$ (a diferencia de $\delta(S)$, que denota el conjunto de enlaces con un extremos en S_1 y otro en $V \setminus S_1$).

- (S_1, S_2) : conjunto de enlaces con un extremo en S_1 y otro en S_2 :
- $A(S_1, S_2)$: conjunto de arcos que salen de S_1 y entran en S_2 .
- $E(S_1, S_2)$: conjunto de aristas entre S_1 y S_2 .

Diremos que un vértice es par (impar) cuando es incidente con un número par (impar) de enlaces, y un grafo es par cuando lo son todos sus vértices.

Cada solución del MGRP está representada por un vector $y = (y_e : e \in E \cup A) \in \mathbb{R}^{|E|+|A|}$, donde y_e es el número de veces que el enlace e es recorrido en la solución y . Esta representación de las soluciones está basada en la caracterización de un grafo mixto Euleriano introducida por Ford y Fulkerson (1962): G es Euleriano si y sólo si es conexo, par y equilibrado. Se dice que un grafo G es equilibrado si para cada conjunto de vértices $S \subset V$, la diferencia entre el número de arcos salientes y entrantes en S es menor o igual al número de aristas en $\delta(S)$, es decir, $|A^+(S) - A^-(S)| \leq |E(S)|$. Como en el WGRP, podemos asumir, sin pérdida de generalidad, que todos los vértices son requeridos ($V_R = V$). También podemos suponer que todas las aristas son requeridas ($E_R = E$), ya que podemos sustituir las aristas no requeridas por dos arcos no requeridos, uno en cada dirección. Dados un vector solución y y un conjunto de enlaces H , denotaremos $y(H)$ a la suma de las variables asociadas a los enlaces en H . De esta manera, las soluciones del MGRP en un grafo $G = (V, E, A) = (V_R, E_R, A_R \cup A_{NR})$ serán los vectores $y \in \mathbb{R}^{|E|+|A|}$ que cumplan:

$$y_{ij} \geq 1, \quad \forall (i, j) \in A_R \cup E_R \quad (6.5)$$

$$y(A^+(S)) - y(A^-(S)) \leq y(E(S)), \quad \forall S \subset V \quad (6.6)$$

$$y(A^+(S)) \geq 1, \quad \forall S = \bigcup_{k \in Q} V_k, \quad Q \subset \{1, \dots, p\} \quad (6.7)$$

$$y(\delta(\{i\})) \equiv 0 \pmod{2}, \quad \forall i \in V \quad (6.8)$$

$$y_{ij} \geq 0, \quad \forall (i, j) \in A_{NR} \quad (6.9)$$

$$y_{ij} \text{ entero} \quad \forall (i, j) \in A \cup E_R \quad (6.10)$$

Las desigualdades (6.5) aseguran que todos los enlaces requeridos sean recorridos por la solución, mientras que las desigualdades (6.6), (6.7) y (6.8) hacen que sea equilibrado, conexo y par, respectivamente.

Corberán, Romero y Sanchis (2003) demostraron que la dimensión del poliedro del MGRP con esta formulación es $\dim(\text{MGRP}_{F1}(G)) = |E \cup A| - q + 1$, donde q es el número de componentes conexas del grafo (V, E) . Para definir

6.3. Relación de las nuevas facetas con otros problemas similares 115

la configuración \mathcal{C} asociada a las desigualdades Zigzag en $\text{MGRP}_{F_1}(G)$, necesitamos una partición del conjunto de vértices V en cuatro subconjuntos M^1 , M^2 , M^3 y M^4 cumpliendo (aquí, α_{ij} es el número de enlaces requeridos en $(M^i : M^j)$):

- **Caso par:** Cada conjunto M^1 , M^2 , M^3 y M^4 contiene un número par de vértices R-impares y $\alpha_{12} = \alpha_{23} = \alpha_{34} = \alpha_{14} = 0$.
- **Caso impar:** Cada conjunto M^1 , M^2 , M^3 y M^4 contiene un número impar de vértices R-impares y el grafo inducido por los enlaces requeridos en $G_{\mathcal{C}}$ más dos arcos a_{12} y a_{34} , de M^1 a M^2 y de M^3 a M^4 respectivamente, no es equilibrado.

La configuración \mathcal{C} asociada a las desigualdades Zigzag en $\text{MGRP}_{F_1}(G)$ queda definida por el grafo $G_{\mathcal{C}}$ descrito por la partición anterior y los siguientes coeficientes:

- $c(A(M^1, M^2)) = c(A(M^3, M^4)) = 0$
- $c(A(M^2, M^1)) = c(E(M^1, M^2)) = c(A(M^4, M^3)) = 2$
- $c(M^i, M^j) = 1$ en otro caso

La desigualdad Zigzag para $\text{MGRP}_{F_1}(G)$ es:

$$\begin{aligned} & y(M^1 \cup M^2 : M^3 \cup M^4) + 2y((M^1, M^2) \setminus A(M^1, M^2)) + \\ & + 2y((M^3, M^4) \setminus A(M^3, M^4)) \geq \alpha_{13} + \alpha_{24} + \alpha_{23} + \alpha_{14} + \\ & + 2|(M^1 : M^2)_R \setminus A(M^1 : M^2)| + 2|(M^3 : M^4)_R \setminus A(M^3 : M^4)| + 2 \quad (6.11) \end{aligned}$$

La formulación F_1 puede expresarse también en términos de *semitours* (Corberán et al. 2003, 2005bb). Dado un tour del $\text{MGRP}(G)$ representado mediante un vector y , llamamos semitour z a la familia de enlaces resultante de eliminar de dicho tour una copia de cada enlace requerido. Este semitour estará representado por el vector z resultante de restar a y el vector de incidencia y^R de los enlaces requeridos de G , es decir, $y = z + y^R$. De esta manera, la desigualdad Zigzag (6.11) puede escribirse como $F(z + y^R) \geq F(y^R) + 2$, con lo que dicha desigualdad queda, en términos de semitours, como:

$$F(z) \geq 2 \quad (6.12)$$

Teorema 6.3.3 *Las desigualdades Zigzag (6.11) son válidas para $\text{MGRP}_{F_1}(G)$.*

Dem: Si y^R es el vector de incidencia de los enlaces requeridos, $F(y^R) = c_0 - 2$.

En el caso par, el grafo representado por dicho vector no es conexo, ya que no existen enlaces requeridos en la cortadura $\delta(M^1 \cup M^3)$, por lo que debemos recorrer al menos dos veces (ida y vuelta) enlaces de dicha cortadura. Además, el grafo inducido por los enlaces requeridos es par, por lo que sólo podemos añadirle ciclos para mantener la paridad de los vértices. Es fácil ver que no existe ningún ciclo en el grafo configuración que cruce la cortadura y tenga un c -coste menor que dos, por lo que la desigualdad Zigzag se cumple.

En el caso impar, al vector de incidencia y^R hemos de añadirle un matching sobre los cuatro vértices, ya que todos son R -impares. El único matching con c -coste menor que dos es el formado por los arcos a_{12} y a_{34} , que tienen c -coste 0. Pero si añadimos estos arcos, el grafo no será equilibrado, por lo que habrá que añadir más enlaces cruzando la cortadura para hacerlo equilibrado. Como ahora el grafo ya es par todo lo que añadamos ha de formar ciclos, así que, al igual que en el caso par, la desigualdad sumará al menos dos unidades más, con lo que se satisfará. \blacklozenge

Teorema 6.3.4 *Las desigualdades Zigzag (6.11) definen faceta de $MGRP_{F_1}(G)$ si existen, al menos, dos arcos $a_{12} \in A(M^1, M^2)$ y $a_{34} \in A(M^3, M^4)$ y otros dos enlaces definiendo un zigzag M^1 - M^2 - M^3 - M^4 - M^1 y se cumple:*

- para el caso par: los enlaces requeridos de G_C inducen un grafo equilibrado.
- para el caso impar: existen dos aristas requeridas $e_{13} \in (M^1, M^3)$ y $e_{24} \in (M^2, M^4)$ y los enlaces requeridos restantes en G_C inducen un grafo equilibrado.

Dem: Veamos primero el caso impar. La dimensión de $MGRP(G_C)$ es $|E^C \cup A^C| - q + 1$, donde q es el número de componentes conexas de (V^C, E^C) , que puede ser 1 (si hay aristas en $\delta(M^1 \cup M^3)$) o 2 (si no hay aristas en dicha cortadura). Necesitaremos entonces $|E^C \cup A^C|$ o $|E^C \cup A^C| - 1$ tours linealmente independiente que cumplan la restricción con igualdad, dependiendo de si hay o no aristas en la cortadura.

En primer lugar, seleccionaremos una serie de enlaces. Sean e_{13} y e_{24} las aristas requeridas y a_{12} y a_{34} los arcos mencionados en el enunciado del teorema. Si existe alguna arista entre M^1 y M^2 o M^2 y M^1 , seleccionaremos dichas aristas en vez de los arcos, pero, por conveniencia, utilizaremos la misma notación para designarlos. Sean l_{23} y l_{14} los otros dos enlaces que definen el zigzag (pueden ser arcos o aristas, pero, en caso de existir aristas, seleccionaremos éstas). Llamaremos y^d al vector de incidencia de los enlaces requeridos excepto e_{13} y e_{24} . El grafo asociado a este vector es par y equilibrado (aunque podría no ser conexo si no existen aristas en la cortadura $\delta(M^1 \cup M^3)$). Dado un enlace $l \in E^C \cup A^C$, denotaremos $y(l)$ al vector de $\mathbb{R}^{|E^C|+|A^C|}$ con el valor 1 en la componente asociada al enlace l y 0 en las demás. De esta manera, construimos los siguientes tours:

6.3. Relación de las nuevas facetas con otros problemas similares 117

- Para cada arco a de M^1 a M^2 excepto a_{12} , construimos el tour $y^d + y(a) + y(l_{23}) + y(e_{24}) + y(e_{13}) + y(e_{24})$.
- Para cada arco a de M^2 a M^1 , construimos el tour $y^d + y(a) + y(a_{34}) + y(e_{13}) + y(e_{24})$.
- Para cada arista e entre M^1 y M^2 , construimos el tour $y^d + y(e) + y(a_{34}) + y(e_{13}) + y(e_{24})$.
- Para cada enlace $l \in (M^1, M^3) \setminus \{e_{13}\}$, construimos el tour $y^d + y(l) + y(l_{23}) + y(a_{34}) + y(e_{13}) + y(e_{24})$.
- Para cada arco a de M^1 a M^4 , construimos el tour $y^d + y(a) + y(l_{23}) + y(e_{13}) + y(e_{24})$.
- Para cada arco a de M^4 a M^1 (excepto, si es un arco, l_{14}), construimos el tour $y^d + y(a) + y(a_{12}) + y(e_{13}) + y(e_{13}) + y(e_{24})$.
- Para cada arista e entre M^1 y M^4 (excepto, si es una arista, l_{14}), construimos el tour $y^d + y(e) + y(a_{12}) + y(e_{13}) + y(e_{13}) + y(e_{24})$.

Para el resto de enlaces excepto los mencionados al principio de la demostración, podemos construir un tour de manera análoga a los ya descritos. De esta manera hemos obtenido $|E^C \cup A^C| - 6$ tours. Utilizando únicamente los enlaces $e_{13}, e_{24}, a_{12}, a_{34}, l_{23}$ y l_{14} , podemos construir otros 5 tours:

- $y^d + y(a_{12}) + y(a_{34}) + y(e_{13}) + y(e_{24}) + 2y(l_{23})$
- $y^d + y(a_{12}) + 2y(e_{13}) + y(e_{24}) + y(l_{14})$
- $y^d + y(a_{34}) + 2y(e_{13}) + y(e_{24}) + y(l_{23})$
- $y^d + y(a_{34}) + y(e_{13}) + 2y(e_{24}) + y(l_{14})$
- $y^d + y(a_{12}) + y(e_{13}) + 2y(e_{24}) + y(l_{23})$

Si existe alguna arista en la cortadura $\delta(M^1 \cup M^3)$, podemos también construir uno de estos dos tours:

- Si hay alguna arista en (M^1, M^4) o (M^2, M^3) , construimos el tour $y^d + y(e_{13}) + y(e_{24}) + y(l_{23}) + y(l_{14})$.
- Si no hay aristas en (M^1, M^4) ni (M^2, M^3) , habrá alguna en (M^1, M^2) o (M^3, M^4) , y construiremos el tour $y^d + y(e_{13}) + y(e_{24}) + y(a_{12}) + y(a_{34})$ (recordemos que, en este caso, a_{12} y a_{34} pueden ser aristas).

Tenemos, por tanto, los tours que necesitamos, y es fácil ver que todos cumplen la restricción Zigzag con igualdad. Para ver que son linealmente independientes, los podemos representar en forma de matriz, restando el vector y^d a todos ellos, con lo que obtenemos:

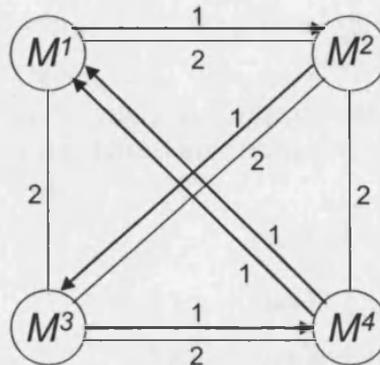


Figura 6.27: Instancia con Zigzag válida para F2 pero no F1

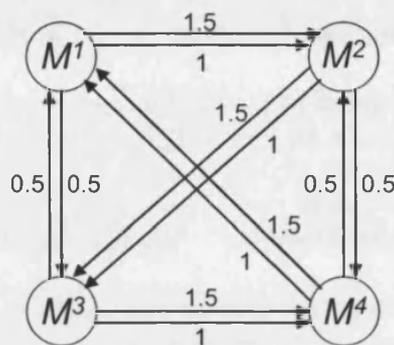


Figura 6.28: Solución fraccionaria (formulación F2)

Debemos notar que, a pesar de que las desigualdades Zigzag son válidas y definen faceta tanto para $MGRP_{F2}(G)$ como para $MGRP_{F1}(G)$, no existe una correspondencia uno a uno entre estas facetas. Consideremos la instancia que se observa en la figura 6.27, donde todos los enlaces son requeridos. Empleando un algoritmo de planos de corte (Corberán, Mota y Sanchis, 2005bb) basado en la formulación $F2$, obtenemos una solución fraccionaria (ver figura 6.28) que viola la desigualdad Zigzag de la figura 6.29. Sin embargo, la desigualdad Zigzag que cortaría a dicha solución con la formulación $F1$ (figura 6.30) no es válida. Esto se debe a que, si añadimos dos arcos de M^1 a M^2 y de M^3 a M^4 , el grafo resultante sí es equilibrado. En cualquier caso, esta solución (figura 6.31) no podría aparecer en un algoritmo de planos de corte con la formulación $F1$, puesto que es combinación lineal de dos tours del MGRP (ver figura 6.32).

6.3. Relación de las nuevas facetas con otros problemas similares 121

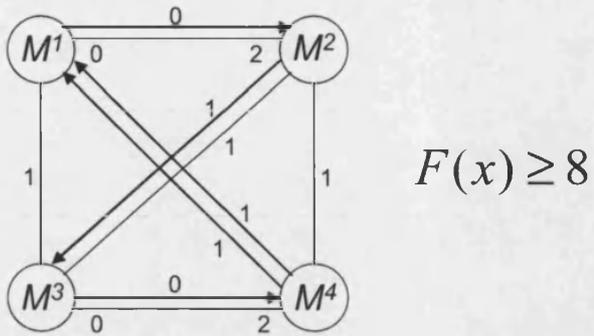


Figura 6.29: Desigualdad Zigzag válida para F2

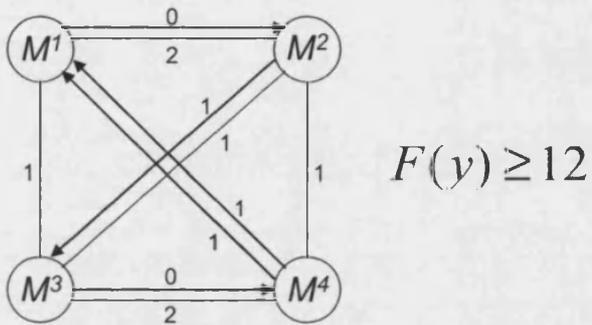


Figura 6.30: Desigualdad Zigzag no válida para F1

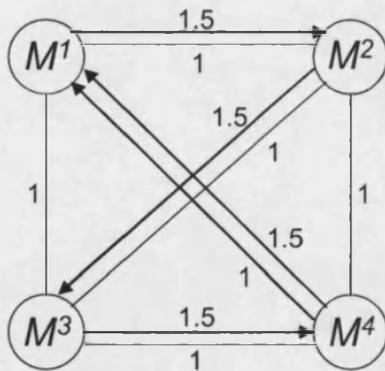


Figura 6.31: Solución fraccionaria (formulación F1)

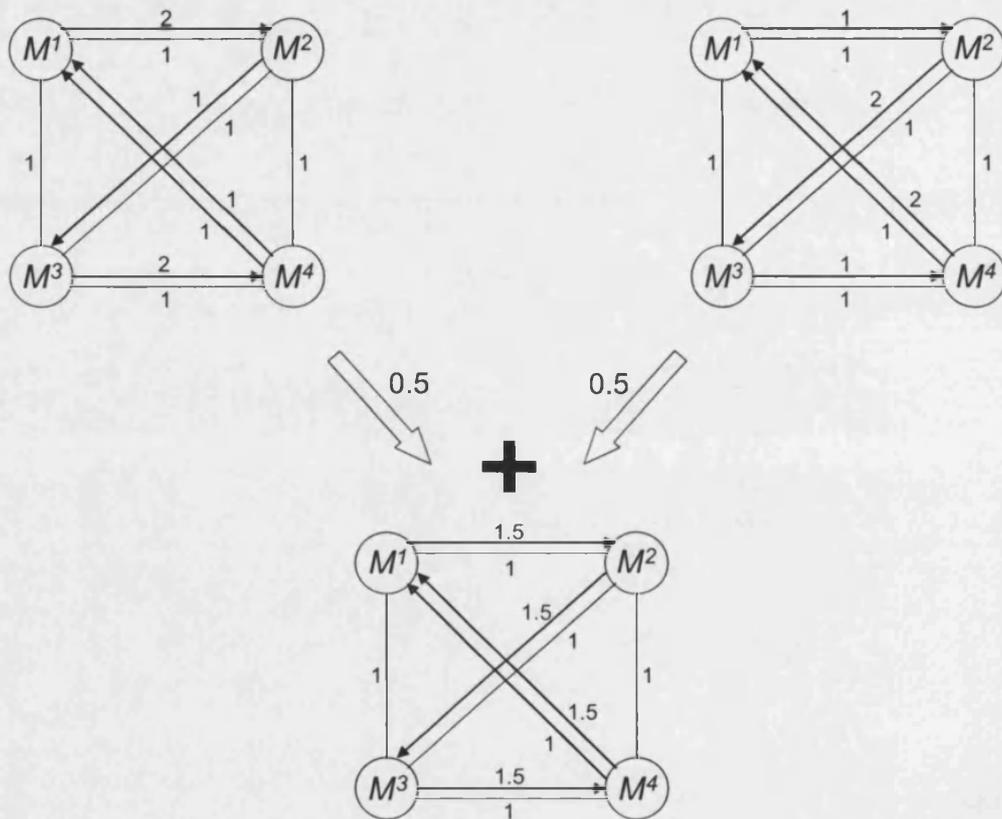


Figura 6.32: Zigzag para F1 como combinación lineal

Capítulo 7

Branch And Cut

Los algoritmos de tipo *Branch & Cut* (ramificación y corte) fueron empleados por primera vez en un Problema de Rutas por Padberg y Rinaldi (1990). Consisten, básicamente, en un árbol de enumeración (como en un Branch & Bound), en el que, en cada nodo, se aplica un algoritmo de *planos de corte* para resolver el subproblema ahí definido. Éste emplea una serie de *algoritmos de separación* para identificar desigualdades válidas violadas por las soluciones no factibles, llamadas *cortes*.

7.1. Problema lineal inicial

El problema lineal (PL) inicial que empleamos en el nodo raíz del árbol es el siguiente:

$$\text{Min } \sum_{(i,j) \in E} (c_{ij}x_{ij} + c_{ji}x_{ji}) \tag{7.1}$$

$$\text{s.a. : } \quad x_{ij} + x_{ji} \geq 1 \quad \forall (i,j) \in E_R \tag{7.2}$$

$$\sum_{(i,j) \in \delta(i)} (x_{ij} - x_{ji}) = 0 \quad \forall i \in V \tag{7.3}$$

$$\sum_{(i,j) \in V_k, j \in V \setminus V_k} x_{ij} \geq 1 \quad \forall k \in \{1, \dots, p\} \tag{7.4}$$

$$x(\delta(S)) \geq |\delta_R(S)| + 1 \quad \forall S \in \mathcal{S} \tag{7.5}$$

$$x_{ij}, x_{ji} \geq 0 \quad \forall (i,j) \in E \tag{7.6}$$

Con respecto a la formulación lineal entera presentada en el capítulo 3, hemos limitado el número de restricciones de conectividad, incluyendo sólo las correspondientes a cada componente R-conexa del grafo (7.4) y hemos eliminado la restricción de integridad sobre las variables. Además, hemos incluido un conjunto de restricciones de cortaduras R-impares seleccionado de la siguiente forma: sea $V_{R-imp} \subseteq V$ el conjunto de vértices R-impares en V . Entonces, $G(V_{R-imp})$ es el grafo inducido por los vértices R-impares. Sea S_i una componente conexa de dicho grafo. Incluiremos S_i en el conjunto \mathcal{S} si S_i contiene un número impar de vértices R-impares, ya que, en ese caso, $\delta(S_i)$ será una cortadura R-impar. También incorporaremos a \mathcal{S} los vértices R-impares de V . Así, incluimos en la formulación las desigualdades de cortaduras R-impares generadas a partir de los conjuntos de vértices incluidos en \mathcal{S} .

Es importante observar que una solución de esta relajación, aún siendo los valores de todas las variables enteras, no será necesariamente una solución factible del problema original, ya que puede inducir un grafo no conexo.

7.2. Cota superior inicial

Una de las claves para la eficiencia de un algoritmo Branch & Cut es disponer de una buena cota superior que nos permita saturar rápidamente el mayor número de nodos posibles. A menudo, alcanzar una solución factible únicamente mediante la ramificación resulta un proceso muy lento, así que hemos optado por el uso de algoritmos heurísticos y metaheurísticos para obtener una cota superior inicial.

Antes de iniciar el algoritmo Branch & Cut, ejecutamos dos algoritmos heurísticos, quedándonos con la mejor de las dos soluciones obtenidas. Los algoritmos que hemos utilizado son los presentados en Benavent et al. (2005). Estos algoritmos proporcionan una cota superior bastante útil en las instancias de tamaños moderados. Sin embargo, en las más grandes el *gap* entre la cota inferior del nodo raíz y la cota superior proporcionada por estos heurísticos resulta demasiado grande para que sea de utilidad. Por ello, hemos utilizado un algoritmo de Búsqueda Dispersa, o Scatter Search (Benavent et al., 2005), que sólo ejecutaremos en el caso en que el *gap* entre la cota superior y la cota inferior en el nodo raíz sea superior al 1%. Este algoritmo metaheurístico consume bastante más tiempo que los heurísticos anteriores, por lo que no lo hace adecuado para instancias pequeñas, donde la mejora de la cota superior sería poco significativa. En las instancias más grandes, sin embargo, el tiempo de ejecución del algoritmo de Búsqueda Dispersa no es tan grande en relación al tiempo total empleado por el Branch & Cut, y la mejora en la cota superior resulta considerable.

7.3. Tailing-off

Durante la exploración del árbol de Branch & Cut, un nodo puede terminar de ser estudiado por varias razones: se ha saturado por haber llegado a una solución factible o porque el valor del PL resuelto en él supera la cota superior; también si los algoritmos de separación no son capaces de hallar ningún corte nuevo. En este último caso, se procede a la ramificación, creándose dos nuevos nodos para explorar. Sin embargo, mientras los algoritmos sigan identificando restricciones violadas e incorporándolas al PL en ese nodo, y la solución del problema lineal no sobrepase la cota superior, el algoritmo continuará ejecutando las rutinas de separación y resolviendo problemas lineales. A menudo, este proceso se alarga mucho sin que el valor de la función objetivo varíe apenas. En estos casos, puede ser más conveniente proceder a la ramificación para agilizar el proceso global.

Para evitar que el algoritmo pase demasiado tiempo explorando un mismo nodo e incorporando restricciones que no hacen variar apenas el valor de la cota inferior, hemos introducido un criterio, o *tailing-off*, para determinar cuándo esta variación es poco significativa y, por tanto, proceder a ramificar.

Sea f_i el valor óptimo del i -ésimo PL resuelto en ese nodo. Si $\frac{f_i - f_{i-5}}{f_i} \times 100 < 0,0005$, con $i \geq 5$, entonces ramificaremos y pasaremos a explorar otro nodo del árbol. En caso contrario, seguiremos generando cortes mediante los algoritmos de separación. Es decir, ramificamos si la variación de la función objetivo en las 5 últimas iteraciones del algoritmo de planos de corte ha sido menor del 0,0005 %.

Para la fase de separación en el nodo raíz, hemos elegido un porcentaje un poco menor, 0,0001 %. Creemos que es conveniente apurar al máximo la separación en el nodo raíz, ya que el valor de la función objetivo en este nodo servirá de cota inferior para todos los demás nodos, así que vale la pena invertir un poco más de tiempo en mejorar esta cota.

7.4. Ramificación

Una vez el algoritmo ha acabado de buscar cortes en un nodo sin haberlo saturado, bien porque no encuentra ninguno violado, o porque se ha cumplido el criterio para el "tailing-off", hemos de dividir el conjunto de soluciones del nodo actual. Para ello, hemos probado con diferentes estrategias.

La estrategia de ramificación por restricciones intenta aprovechar las características propias del problema para la creación de los subproblemas. Si x^* es

la solución del último problema lineal resuelto en el nodo actual, escogemos una restricción $ax \leq b$ de dicho problema de manera que $b^* = ax^*$ sea un valor no entero. Entonces, generamos dos subproblemas añadiendo a cada uno de ellos las restricciones $ax \leq \lfloor b^* \rfloor$ y $ax \geq \lceil b^* \rceil$ respectivamente. De esta manera conseguimos cortar la solución fraccionaria del último problema lineal.

En nuestra implementación, hemos decidido utilizar restricciones de conectividad y de R-imparidad para la ramificación. En primer lugar, buscamos, entre las restricciones de conectividad $ax \geq 1$ presentes en la relajación inicial tales que $b^* = ax^*$ no toma un valor entero, aquélla para la que $\min\{|b^* - \lfloor b^* \rfloor|, |b^* - \lceil b^* \rceil|\}$ sea máximo. Si no hallamos ninguna restricción de conectividad para la que b^* sea no entero, buscamos de la misma manera entre las restricciones de R-imparidad presentes en la relajación inicial. De esta manera, ramificaremos utilizando la restricción seleccionada. De no encontrar ninguna restricción de conectividad o R-imparidad cumpliendo esta condición, hemos de utilizar una estrategia alternativa.

Como posibles estrategias a utilizar en el caso en que no se encuentren restricciones adecuadas, hemos probado con tres estrategias de ramificación por variables. Una estrategia de ramificación por variables consiste simplemente en escoger una variable x_e que tome un valor no entero en la solución x^* del nodo actual y generar dos nuevos subproblemas añadiendo las restricciones $x_e \leq \lfloor x_e^* \rfloor$ y $x_e \geq \lceil x_e^* \rceil$ respectivamente. Para la elección de la variable, hemos probado con 3 estrategias diferentes implementadas en Cplex 9.0. La primera consiste en seleccionar la variable cuya parte fraccionaria en la solución esté más próxima a 0.5, i.e., aquélla tal que $\min\{|x_e^* - \lfloor x_e^* \rfloor|, |x_e^* - \lceil x_e^* \rceil|\}$ sea máximo. La segunda posibilidad es utilizar la estrategia llamada *strong branching*. Esta estrategia, introducida por primera vez por Applegate, Bixby, Chvátal y Cook (1995) consiste en escoger un conjunto de variables con valor no entero en la solución y resolver los problemas lineales que resultarían al ramificar por cada una de dichas variables. Estos problemas no se resuelven necesariamente de forma óptima, sino que basta con realizar algunas iteraciones del algoritmo símplex para obtener una estimación del nuevo valor de la función objetivo. De esta manera, se puede estimar cuál sería el incremento en la función objetivo en caso de ramificar por cada una de las posibles variables, eligiendo aquélla para la cual dicho incremento sea mayor. Por último, hemos probado dejando total libertad de elección al algoritmo de ramificación de Cplex.

También hemos probado a utilizar exclusivamente la estrategia de ramificación *strong branching*, en vez de la ramificación por restricciones.

7.5. Variaciones menos densas de algunas restricciones

Uno de los puntos críticos para poder resolver óptimamente instancias de gran tamaño es el uso de la memoria RAM disponible. Conforme se van introduciendo más cortes al problema y el tamaño del árbol del B&C crece, la cantidad de memoria utilizada para almacenar toda esta información aumenta considerablemente, pudiendo llegar a agotarse completamente el espacio disponible, lo que nos impediría continuar con la resolución del problema. Es por ello que resulta de suma importancia guardar las restricciones que los algoritmos de separación van encontrando de la manera más eficiente posible. Una de las formas de conseguir esto es expresarlas de manera que el número de coeficientes no nulos sea lo más pequeño posible. Para ello, emplearemos el hecho de que, dado un subconjunto cualquiera de nodos $S \subset V$, el número de veces que un tour entra en él es igual al número de veces que sale de él, es decir:

$$\sum_{i \in S, j \in V \setminus S} x_{ij} = \sum_{i \in S, j \in V \setminus S} x_{ji}$$

Esta igualdad es fácilmente deducible a partir de las ecuaciones de simetría en cada vértice.

7.5.1. Desigualdades R-impares

Es fácil ver que las restricciones de cortes R-impares

$$\sum_{(i,j) \in \delta(S)} (x_{ij} + x_{ji}) \geq |\delta_R(S)| + 1 \quad \forall S \quad R - \text{impar}$$

pueden escribirse de la siguiente forma:

$$\sum_{(i,j): i \in S, j \in V \setminus S} x_{ij} \geq \frac{|\delta_R(S)| + 1}{2} \quad \forall S \quad R - \text{impar}$$

que tiene exactamente la mitad de elementos no nulos que la versión original.

7.5.2. Desigualdades K-C y K-C₀₂

Si consideramos las ecuaciones de simetría en los conjuntos M_i con i impar para sustituir las variables correspondientes a las unidades que salen de dichos M_i en las desigualdades K-C, los coeficientes que quedan, después de dividir toda la desigualdad por dos, es fácil ver que son (figura 7.1):

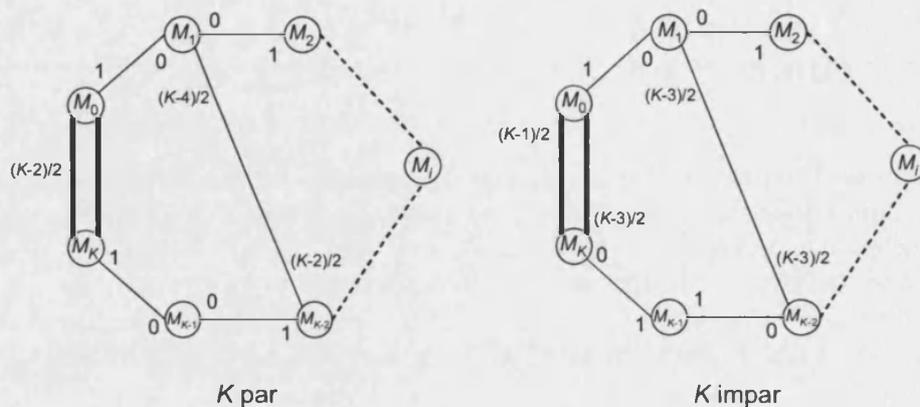


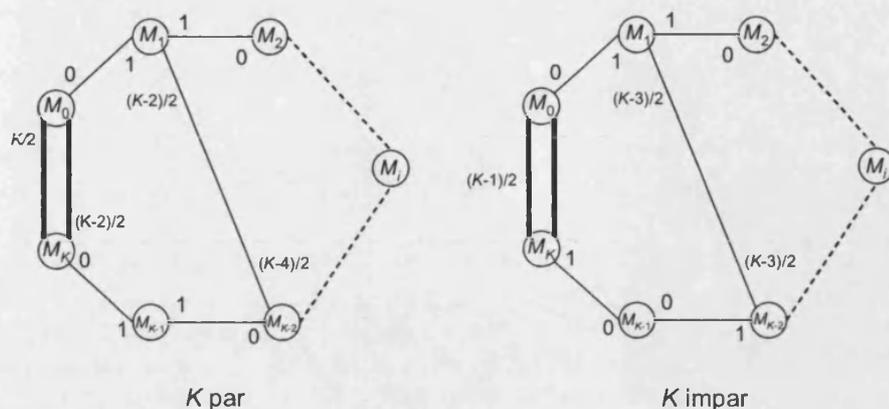
Figura 7.1: Coeficientes modificados para K-C

$$\begin{aligned}
 \blacksquare c(M_i, M_j) &= \begin{cases} \frac{|j-i|}{2} & \text{si } i \text{ y } j \text{ son ambos pares o ambos impares} \\ \frac{|j-i|-1}{2} & \text{si } i \text{ es impar y } j \text{ es par} \\ \frac{|j-i|+1}{2} & \text{si } i \text{ es par y } j \text{ es impar} \end{cases} \\
 \blacksquare c(M_0, M_K) &= \begin{cases} \frac{K-2}{2} & \text{si } K \text{ es par} \\ \frac{K-1}{2} & \text{si } K \text{ es impar} \end{cases} \\
 \blacksquare c(M_K, M_0) &= \begin{cases} \frac{K-2}{2} & \text{si } K \text{ es par} \\ \frac{K-3}{2} & \text{si } K \text{ es impar} \end{cases} \\
 \blacksquare c_0 &= K - 1 + \frac{K-2}{2} |(M_0 : M_K)_R|
 \end{aligned}$$

Como se puede apreciar en la figura 7.1, un gran número de los coeficientes son ahora 0's, por lo que una desigualdad K-C en esta forma es menos densa que en su forma general.

Para las desigualdades K-C₀₂ podemos hacer lo mismo, sólo que sustituiremos las variables que salen de los nodos M_i con i par. De esta forma, obtenemos los siguientes coeficientes (figura 7.2):

$$\blacksquare c(M_i, M_j) = \begin{cases} \frac{|i-j|}{2} & \text{si } i \text{ y } j \text{ son ambos pares o ambos impares} \\ \frac{|i-j|+1}{2} & \text{si } i \text{ es impar y } j \text{ es par} \\ \frac{|i-j|-1}{2} & \text{si } i \text{ es par y } j \text{ es impar} \end{cases}$$

Figura 7.2: Coeficientes modificados para $K-C_{02}$

$$i \neq 0, j \neq 0$$

$$\blacksquare c(M_0, M_j) = \begin{cases} \frac{j+1}{2} & \text{si } j \text{ es impar} \\ \frac{j}{2} + 1 & \text{si } j \text{ es par} \end{cases} \quad j \neq K$$

$$\blacksquare c(M_j, M_0) = \begin{cases} \frac{j-1}{2} & \text{si } j \text{ es impar} \\ \frac{j}{2} - 1 & \text{si } j \text{ es par} \end{cases} \quad j \neq K$$

$$\blacksquare c(M_0, M_K) = \begin{cases} \frac{K}{2} & \text{si } K \text{ es par} \\ \frac{K-1}{2} & \text{si } K \text{ es impar} \end{cases}$$

$$\blacksquare c(M_K, M_0) = \begin{cases} \frac{K-2}{2} & \text{si } K \text{ es par} \\ \frac{K-1}{2} & \text{si } K \text{ es impar} \end{cases}$$

$$\blacksquare c_0 = K - 1 + \frac{K-1}{2} |(M_0 : M_K)_R|$$

Al igual que ocurría con las $K-C$ estándar, esta nueva versión de las desigualdades $K-C_{02}$ es más útil desde un punto de vista computacional.

7.6. Cota superior artificial y reinicio

Tras una serie de pruebas computacionales, hemos podido comprobar que, en algunas de las instancias más grandes, el *gap* entre la cota inferior y la cota

superior en el nodo raíz es muy grande. Esto hace que sea muy difícil saturar nodos y, dado que el valor de la función objetivo aumenta muy despacio en sucesivas iteraciones, imposibilita alcanzar una solución óptima del problema. Observando los resultados correspondientes a instancias sí resueltas de tamaños similares, hemos observado que, normalmente, la cota inferior en ese momento suele estar más próxima a la solución óptima del problema que la cota superior.

Para solucionar este problema, hemos decidido introducir una cota superior artificial en aquellos casos en los que el *gap* en el nodo raíz sea “muy grande”. Esperamos así saturar nodos del árbol que conducirían a soluciones peores que la óptima. Esta estrategia tiene el inconveniente de que, si escogemos una cota superior ficticia demasiado baja, puede que ésta sea inferior incluso a la solución óptima, con lo que no encontraríamos el óptimo del problema. Por ello, hemos introducido un mecanismo de reinicio, de manera que, si el algoritmo finaliza la búsqueda sin hallar una solución factible por debajo de la cota ficticia, reinicie de nuevo la búsqueda con una nueva cota superior artificial ligeramente superior a la anterior.

Si fuera éste el caso, para aprovechar la información obtenida en el primer árbol de búsqueda, guardaremos algunos de los cortes hallados durante la ejecución del Branch & Cut en la primera iteración. En concreto, si x^* es la última solución fraccionaria obtenida en el nodo raíz de la primera iteración del Branch & Cut, guardaremos, de entre todos los cortes $ax \geq b$ hallados hasta el momento, aquéllos *más ajustados* en x^* , es decir, aquéllos que cumplan $ax^* \leq b + \epsilon$. Al ejecutar la segunda iteración del Branch & Cut, introduciremos dichos cortes en el PL inicial, además de las restricciones que definían el PL de la primera iteración.

En el caso de que la segunda iteración finalizara también sin haber hallado una solución posible, volvemos a reiniciar, utilizando esta vez la verdadera cota superior.

En el siguiente esquema podemos ver con más claridad la estructura del algoritmo de reinicio:

1. Ejecutar heurísticos. Sea UB el valor de la mejor solución encontrada.
2. $It = 1$.
3. Construir el PL inicial.
4. Si $It \geq 2 \rightarrow$ añadir los cortes almacenados tales que $ax^0 < b + \epsilon$ y las restricciones $\sum_{(i,j) \in E} (c_{ij}x_{ij} + c_{ji}x_{ji}) \geq LB$ y $\sum_{(i,j) \in E} (c_{ij}x_{ij} + c_{ji}x_{ji}) \leq UB_a$.
5. Aplicar el algoritmo de planos de corte en el nodo raíz.

6. Sea x^0 la última solución fraccionaria del nodo raíz y LB_0 su valor.
7. Si $It=1$:
 - 7.1. Si $UB > K_1 \times LB_0 \rightarrow$ ejecutar metaheurístico y actualizar UB .
 - 7.2. Si $UB > K_2 \times LB_0 \rightarrow$ Utilizar cota $UB_a = K_3 \times LB_0$
8. Continuar ejecutando B&C.
9. Si B&C imposible:
 - 9.1. Si $It = 1 \rightarrow UB_a = K_4 \times LB_0, LB = UB_a$.
 - 9.2. Si $It = 2 \rightarrow UB_a = UB$.
 - 9.3. $It++$.
 - 9.4. Volver a 3.
10. Si B&C óptimo o tiempo límite alcanzado \rightarrow FIN.

Los parámetros utilizados son los siguientes:

- $K_1(K_1 > 1)$ es el *gap* a partir del cual se ejecuta el algoritmo metaheurístico para mejorar la cota superior.
- $K_2(K_2 > 1)$ es el *gap* a partir del cual utilizamos una cota superior artificial.
- $K_3(K_3 > 1)$ es el porcentaje respecto de la cota inferior del nodo raíz al que se sitúa la cota superior artificial en la primera iteración.
- $K_4(K_4 > 1)$ es el porcentaje respecto de la cota inferior del nodo raíz al que se sitúa la cota superior artificial en la segunda iteración.

Los valores empleados para los parámetros son $K_1 = 1,01$, $K_2 = 1,015$, $K_3 = 1,005$ y $K_4 = 1,01$.

7.7. Filtrado de restricciones de R-imparidad

En la resolución de instancias de gran tamaño, el número total de cortes introducidos es bastante considerable, pudiendo llegar a ser del orden de decenas de miles. Esto hace que el tamaño de los PL's aumente rápidamente. Además, como todas estas desigualdades que se van generando han de ser mantenidas simultáneamente en memoria, puede agotarse la memoria RAM disponible, produciéndose una ralentización en la ejecución. Algunas instancias pueden no ser resueltas a causa de este problema.

La mayor cantidad de desigualdades halladas por los algoritmos de separación corresponde a restricciones de cortes R-impares, puesto que el algoritmo exacto de separación para estas desigualdades suele encontrar una gran cantidad de desigualdades violadas. Sin embargo, muchas de estas restricciones son similares entre sí, diferenciándose tan sólo en unas pocas aristas de la cortadura. Así pues, hemos decidido incorporar tan solo aquéllas que se diferencien bastante entre sí.

Para ello, generamos mediante el algoritmo exacto de separación el conjunto R de todas las restricciones de cortes R-impares violadas por la solución fraccionaria. De entre ellas, escogemos la más violada, la eliminamos de R y la incorporamos a un nuevo conjunto S . Si k_i es el número de coeficientes no nulos en una restricción i , definimos la *similitud*, $sim(i, j)$, entre dos restricciones i y j como el número de variables con coeficientes no nulos en ambas restricciones (es decir, la mitad del número de aristas comunes a las dos cortaduras definidas por las respectivas restricciones), dividido por $max\{k_i, k_j\}$. A su vez, definimos la similitud de una restricción i a un conjunto de restricciones S como la máxima similitud entre i y las restricciones de dicho conjunto, $sim(i, S) = max\{sim(i, j) : j \in S\}$. Así, en cada paso escogemos aquella restricción de R cuya similitud con S es menor. Si dicha similitud es menor que 0.9, la incorporamos al conjunto S , eliminándola de R . Si no, paramos. El conjunto de restricciones S así obtenido será el que añadiremos al PL correspondiente. En forma abreviada:

1. Generar el conjunto R de desigualdades de R-imparidad violadas.
2. Escoger la desigualdad $i \in R$ más violada. Hacer $S = \{i\}$ y $R = R \setminus \{i\}$.
3. Escoger una desigualdad $i \in R$ tal que $sim(i, S) \leq sim(j, S) \forall j \in R$.
4. Si $sim(i, S) \leq 0,9$, hacer $S = S \cup \{i\}$, $R = R \setminus \{i\}$ y volvemos al paso 3. En otro caso, añadimos S al PL correspondiente y salimos.

7.8. Algoritmos de separación

Para la identificación de desigualdades violadas por soluciones no factibles, hemos implementado algoritmos heurísticos para todos los tipos de desigualdades presentados, así como otros dos algoritmos exactos para la separación de las desigualdades de conectividad y de cortaduras R-impares. Exceptuando el algoritmo correspondiente a las desigualdades Zigzag, que ha sido diseñado explícitamente para este problema, el resto de algoritmos han sido adaptados a partir de otros ya existentes para otros problemas.

Los casos en los que se ha adaptado el algoritmo a nuestro problema tienen una primera fase común, en la que, a partir de la solución no factible x^* , obtenemos un grafo G^* con pesos w en las aristas. Para ello, dada una arista $e = (i, j)$, definimos $w_e = x_{ij}^* + x_{ji}^*$ si e es no requerida y $w_e = x_{ij}^* + x_{ji}^* - 1$ si e es requerida. Al peso total de un conjunto de aristas T lo denotaremos $w(T)$.

A continuación, comentamos los diferentes procedimientos implementados para la separación de las desigualdades utilizadas en la resolución del WGRP.

7.8.1. Desigualdades de conectividad

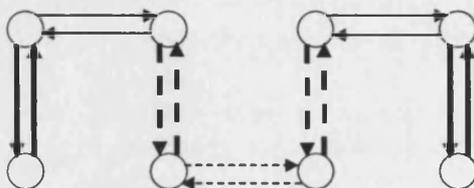
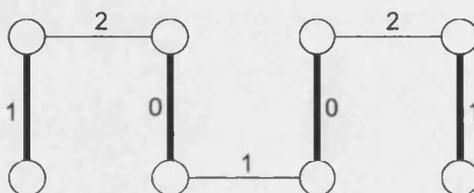
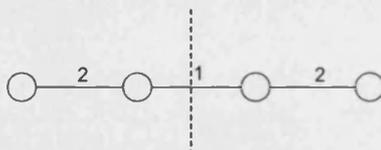
Las restricciones de conectividad pueden ser identificadas de manera exacta en tiempo polinómico. Para ello, construimos un grafo auxiliar comprimiendo las componentes R-conexas de G^* en un solo nodo. Las aristas paralelas resultantes se comprimen en una sola, siendo su nuevo peso la suma de los pesos de las aristas originales. Ahora, basta con hallar una cortadura de peso mínimo en este grafo, lo cual puede hacerse mediante un algoritmo de orden $O(nm + n^2 \log n)$ (Nagamochi, Ono e Ibaraki, 1994). Si la cortadura resultante tiene peso menor que 2, la restricción de conectividad asociada a dicha cortadura estará violada por la solución. En las figuras 7.3 y 7.4 podemos ver un ejemplo de solución fraccionaria y de cortadura hallada por este algoritmo. Las líneas gruesas corresponden a aristas requeridas y las más finas a no requeridas. Cuando una arista es recorrida en la solución con valor 0.5 lo hemos representado mediante un arco discontinuo (a partir de ahora utilizaremos este tipo de representación en los ejemplos).

El algoritmo heurístico que hemos empleado para este tipo de restricciones comienza considerando el grafo resultante de comprimir las componentes R-conexas en G^* y eliminar del grafo aquellas aristas con peso menor o igual que ϵ . Si dicho grafo resulta no ser conexo, consideraremos las cortaduras definidas por sus componentes conexas S_1, S_2, \dots, S_q para comprobar si las correspondientes restricciones de conectividad son violadas o no por la solución. Este algoritmo tiene complejidad $O(|E|)$ y se ejecuta para distintos valores de $\epsilon \geq 0$.

7.8.2. Cortaduras R-impares

Este tipo de desigualdades también puede ser identificado en tiempo polinómico por medio del procedimiento de Padberg y Rao (1982) para encontrar cortaduras impares mínimas.

Sin embargo, este algoritmo consume una gran cantidad de tiempo, por lo que es útil el uso de un algoritmo heurístico más rápido. Al igual que en la sepa-

Solución no factible x^* Grafo G^* Figura 7.3: Solución que viola una restricción de conectividad y grafo G^* Figura 7.4: Grafo comprimido y cortadura de peso ≤ 2

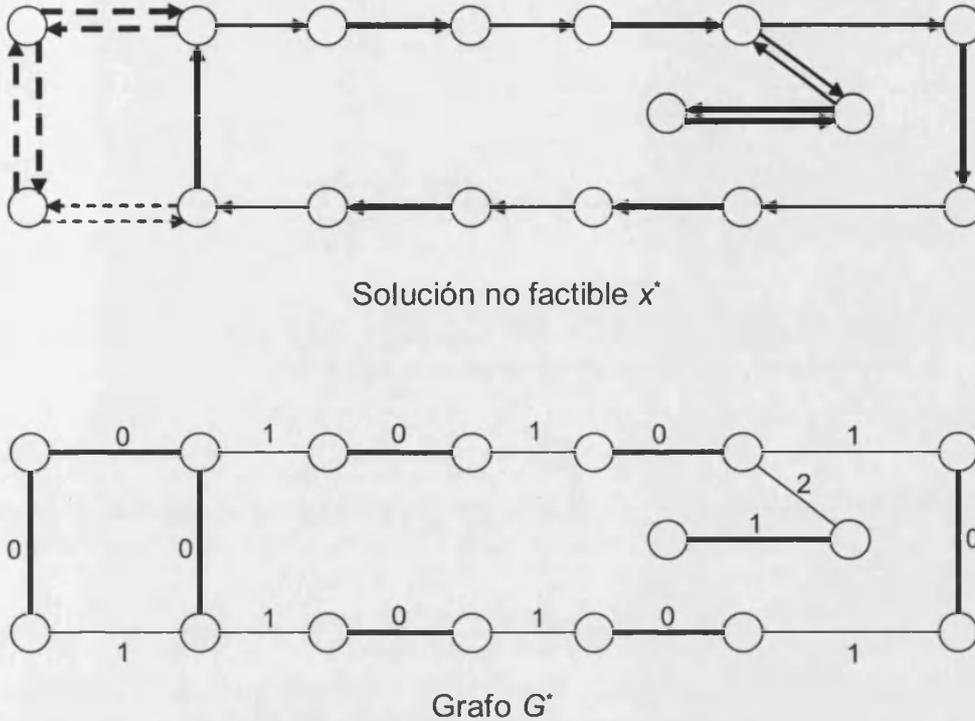


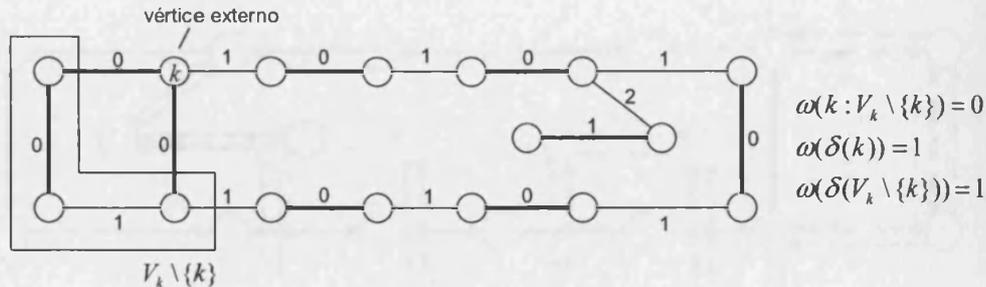
Figura 7.5: Solución que viola una restricción K-C y grafo G^*

ración de las desigualdades de conectividad, trabajaremos con el grafo resultante de eliminar de G^* aquellas aristas con peso menor que ϵ y consideraremos las mismas cortaduras que en ese caso. Si $\delta(S_i)$ es una cortadura R-impar con peso menor que 1, habremos obtenido una desigualdad violada. Este algoritmo puede ejecutarse inicialmente con $\epsilon = 0$ y, en caso de que no se halle ninguna restricción violada (sólo puede ser porque el grafo G^* sea conexo o las cortaduras sean todas R-pares), aplicarse sobre los grafos inducidos por S_1, S_2, \dots, S_q con $\epsilon > 0$, comprobando si las cortaduras resultantes violan la correspondiente restricción de R-imparidad. Cada ejecución del algoritmo tiene complejidad de orden $O(|E|)$.

7.8.3. Desigualdades K-C

El algoritmo heurístico para la identificación de restricciones K-C violadas que se describe a continuación fue presentado por Corberán et al. (2001) para el GRP.

El primer paso consiste en identificar los conjuntos candidatos a formar los

Figura 7.6: Elección de M_0 y M_K

supernodos M_0 y M_K . Diremos que un vértice i en una componente conexa V_i es *exterior* si está conectado a otro vértice de una componente R-conexa V_j distinta por una arista e con peso positivo en G^* . Seleccionaremos una componente R-conexa V_k que contenga, al menos, 2 vértices exteriores.

En primer lugar, tomaremos un vértice exterior $k \in V_k$ R-par. Si $w(k : V_k \setminus \{k\}) = 0$, $w(\delta(k)) \neq 0$ y $w(\delta(V_k \setminus \{k\})) \neq 0$, escogeremos $M_0 = \{k\}$ y $M_K = V_k \setminus \{k\}$ (figura 7.6). En otro caso, probamos con otro vértice exterior R-par de la misma componente. También podemos relajar la condición $w(\delta(V_k \setminus \{k\})) \neq 0$ a $w(\delta(V_k \setminus \{k\})) \leq \epsilon$, lo cual nos da más opciones para encontrar un vértice adecuado, aunque se corre el riesgo de que la desigualdad $K - C$ hallada al final del proceso no esté violada.

En el caso de que no se halle ningún vértice en V_k que satisfaga estas condiciones, volvemos a escoger un vértice exterior k . Sea S_k la componente conexa del subgrafo $G(V_k)$ inducido por las aristas con peso positivo a la que pertenece k (figura 7.7). Si S_k es R-par, $V_k \setminus S_k$ contiene algún vértice exterior y $w(S_k : V_k \setminus S_k) = 0$, escogeremos $M_0 = S_k$ y $M_K = V_k \setminus S_k$. Si no, pasamos a otro nodo exterior. Al igual que antes, podemos utilizar la condición $w(S_k : V_k \setminus S_k) \leq \epsilon$.

Si de esta otra manera tampoco podemos obtener M_0 y M_K , descartaremos la posibilidad de conseguirlos a partir de la componente V_k . Este proceso lo repetimos con cada componente V_k , lo cual nos da la oportunidad de hallar, como máximo, tantas desigualdades K-C como componentes R-conexas en el grafo.

Supongamos que ya hemos hallado dos conjuntos M_0 y M_K . Ahora debemos agrupar el resto de nodos del grafo para formar los conjuntos M_1, M_2, \dots, M_{K-1} . Para ello, comprimiremos en G^* los conjuntos M_0 y M_K , así como las componentes R-conexas (figura 7.8). En este nuevo grafo, hallamos un árbol generador de peso máximo que no contenga la arista $(M_0 : M_K)$ (figura 7.9). Ahora, cal-

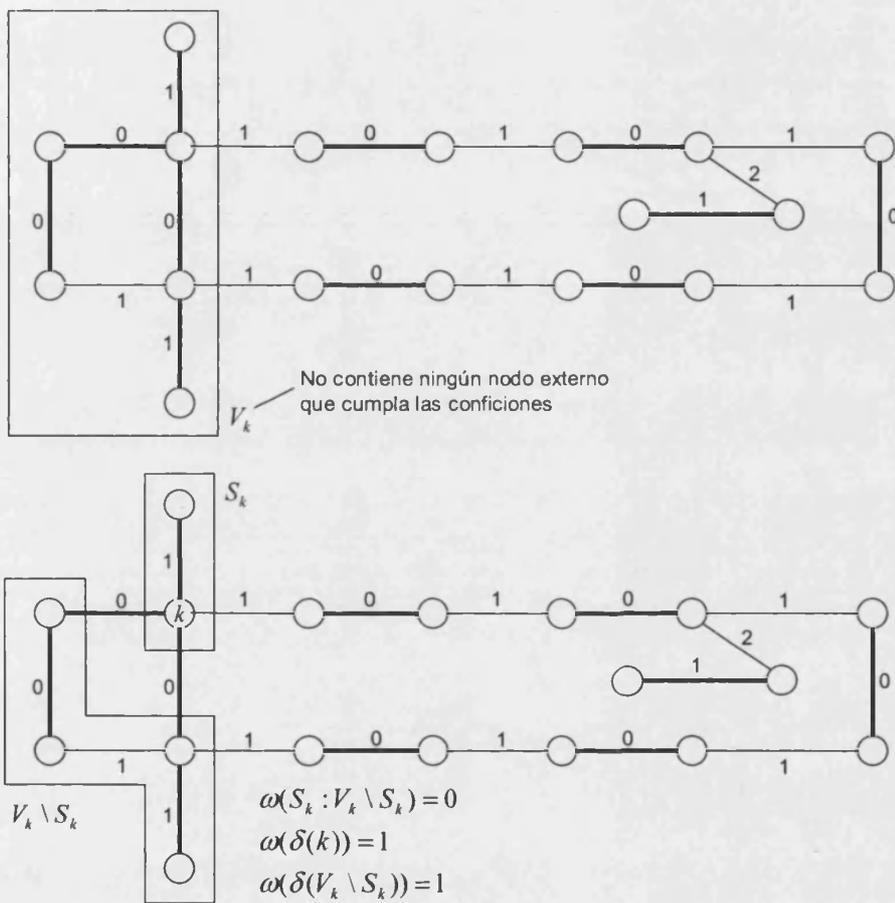


Figura 7.7: Elección de M_0 y M_K alternativa

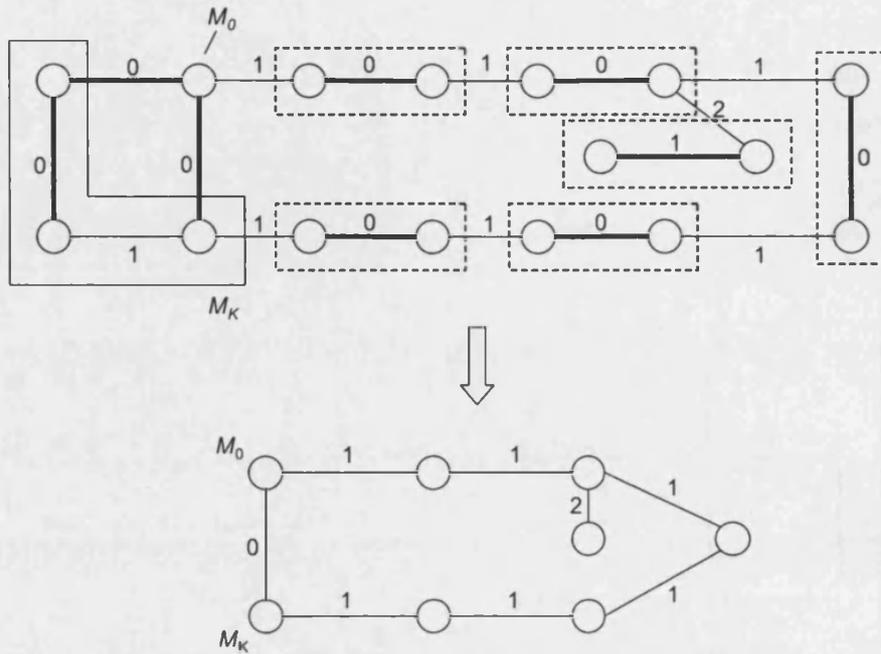


Figura 7.8: Compresión de M_0 , M_K y componentes R-conexas

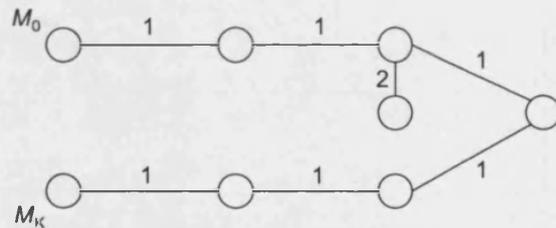


Figura 7.9: Árbol generador de peso máximo en el grafo comprimido

culamos el camino más corto en dicho árbol entre M_0 y M_K . A continuación, comprimiremos sucesivamente cada nodo de grado 1 del árbol (excepto M_0 y M_K) con su nodo adyacente, hasta quedarnos con un camino (i_1, i_2, \dots, i_r) de M_0 a M_K (figura 7.10). Los conjuntos de nodos del grafo original representados por cada i_j serán los conjuntos M_j que utilizaremos.

Una vez tenemos definidos los conjuntos M_0, M_1, \dots, M_K , podemos construir la desigualdad K-C asociada y comprobar si es violada o no por la solución (figura 7.11). En el caso de que esto no fuera así (figura 7.12), aun podemos conseguir una desigualdad K-C violada. Primero probaremos a comprimir pares de

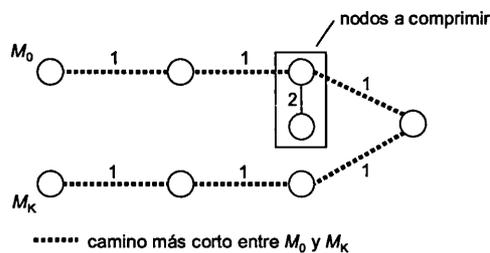


Figura 7.10: Compresión de vértices hasta formar un camino entre M_0 y M_K

conjuntos M_i, M_{i+1} . Si definimos $L_i = w(\delta(M_0 \cup M_1 \cup \dots \cup M_i)) - w(M_0 : M_K)$, este valor representa la cantidad en que disminuiría la parte de la izquierda en la desigualdad K-C evaluada en x^* si unimos los conjuntos M_i y M_{i+1} , $i \neq 0$. A su vez, el término de la derecha disminuye en 2 por cada par de conjuntos que unimos, así que conviene unir todos los pares para los que $L_i > 2$ (siempre y cuando no hagamos K menor que 3) (figura 7.13). De igual manera, si comprimimos los conjuntos M_0 y M_1 , o M_{K-1} y M_K , o ambos a la vez, obtenemos un *ahorro* (disminución en el término de la izquierda de la desigualdad menos la disminución del de la derecha) de

$$\begin{cases} a &= L_0 + 2w(M_1 : M_K) - 2 - |(M_0 : M_K)_R| \\ b &= L_{K-1} + 2w(M_0 : M_{K-1}) - 2 - |(M_0 : M_K)_R| \\ c &= a + b + 2w(M_1 : M_{K-1}) \end{cases}$$

respectivamente. Según sea a , b o c el mayor de estos valores, unimos los correspondientes conjuntos (mientras el nuevo K valga al menos 3) y comprobamos si la restricción K-C resultante está violada o no.

Si de esta manera no hemos conseguido obtener una restricción K-C violada, probaremos a comprimir la configuración inicial de otra manera. Para cada par s, t con $0 \leq s, t \leq K - 3$ y $s + t \leq K - 3$, calculamos el *ahorro* que supondría comprimir los conjuntos M_0, M_1, \dots, M_s y $M_{K-t}, M_{K-t+1}, \dots, M_K$. Escogemos entonces el par s, t que proporcione un mayor ahorro y comprobamos si la desigualdad K-C asociada a la configuración resultante de la compresión es violada. Si no es así, el algoritmo habrá finalizado sin poder hallar ninguna desigualdad K-C violada por la solución actual.

Este algoritmo intenta buscar configuraciones K-C a partir de componentes R-conexas (aunque en la última fase, si la desigualdad obtenida no está violada, junta varias componentes en un sólo conjunto). Hemos utilizado también una segunda versión del algoritmo en la que, inicialmente, cogemos dos componentes R-conexas y las consideramos como una sola para la ejecución del algoritmo.

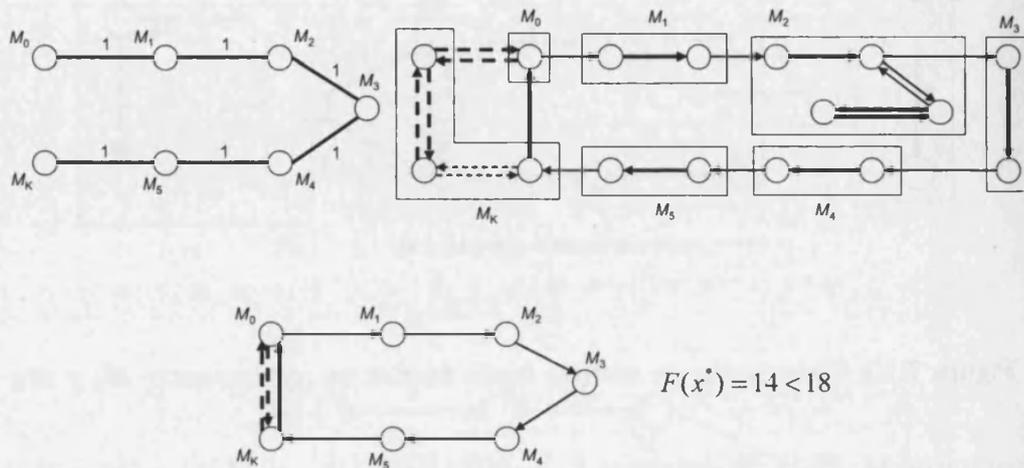


Figura 7.11: Configuración K-C que sí da lugar a una restricción K-C violada

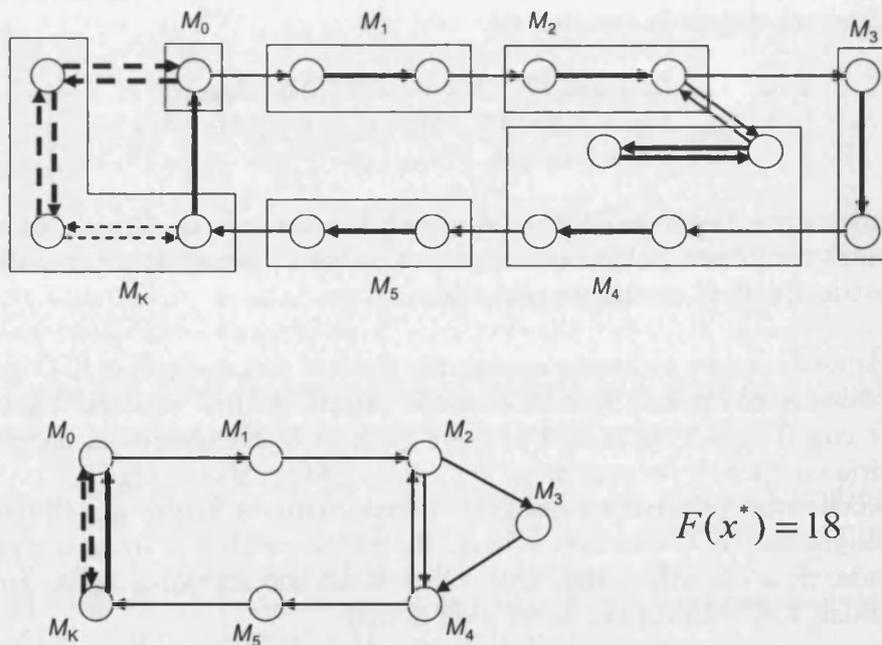


Figura 7.12: Configuración K-C que no da lugar a una restricción K-C violada

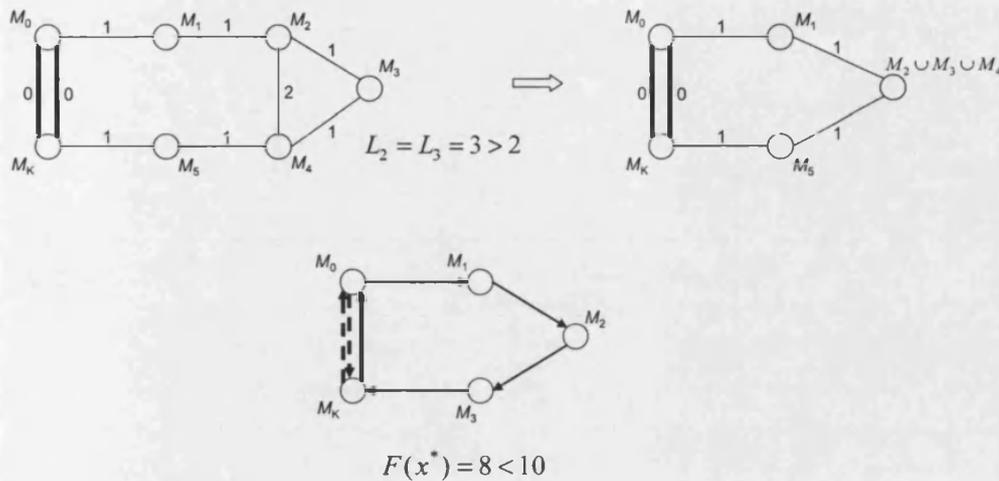


Figura 7.13: Compresión de vértices para obtener una desigualdad K-C violada

Esto lo hacemos para todos los pares posibles de componentes R-conexas.

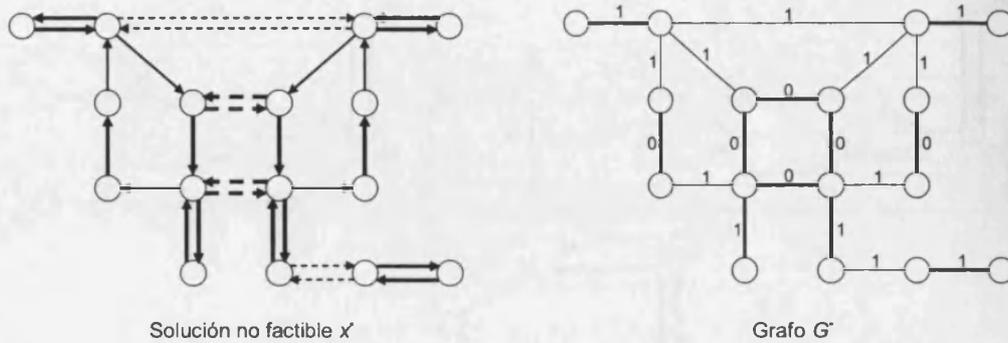
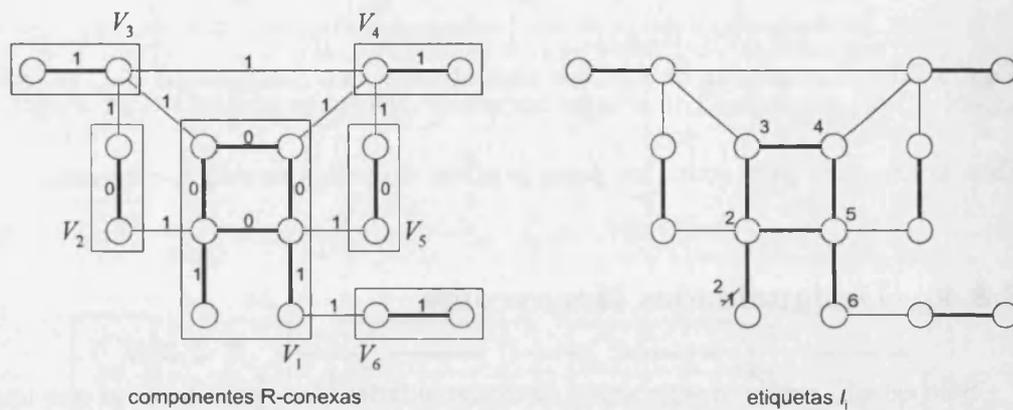
7.8.4. Desigualdades Honeycomb

Sólo estudiaremos la separación de desigualdades Honeycomb en las que una componente R-conexa es particionada en L subconjuntos. El algoritmo heurístico que presentamos a continuación fue propuesto originalmente para el GRP por Corberán et al. (2001).

Consideremos la solución representada en la figura 7.14. En primer lugar, trataremos de hallar los conjuntos B_1, B_2, \dots, B_L a partir de cada uno de los conjuntos R-conexos V_1, V_2, \dots, V_p . Escogemos para ello uno de estos conjuntos, V_r .

Sean u_1, u_2, \dots, u_k los vértices exteriores de V_r . A cada uno de estos vértices u_j le asignaremos la etiqueta $l_j = \arg \max_{i=1, \dots, p, i \neq r} \{w(\{u_j\} : V_i)\}$. Es decir, l_j representará la componente R-conexa unida a u_j por aristas con mayor peso. Al resto de vértices en V_r (los no exteriores) le asignaremos una etiqueta negativa diferente.

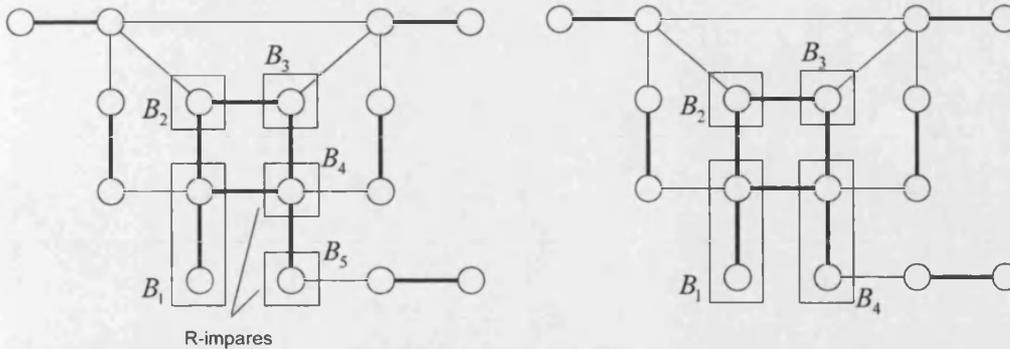
A continuación, iremos escogiendo iterativamente la arista $(u_i, u_j) \in E(V_r)$ de mayor peso aún no seleccionada. Si $l_{min} = \min\{l_i, l_j\}$ y $l_{max} = \max\{l_i, l_j\}$, haremos $l_t = l_{max}$ para todo vértice u_t con $l_t = l_{min}$. Paramos de escoger aristas en V_r cuando todos los vértices de dicha componente tengan etiqueta positiva (figura 7.15). De esta manera podemos construir una partición de V_r en sub-

Figura 7.14: Solución que viola una HC y grafo G^* Figura 7.15: Etiquetado de los vértices de V_r

conjuntos B_1, B_2, \dots, B_t agrupando los vértices que tengan la misma etiqueta.

Si todos estos subconjuntos son R-pares, ya tenemos los conjuntos B_1, B_2, \dots, B_L que buscábamos. Si no, juntamos todos los B_i 's R-impares en un solo conjunto, que será R-par (figura 7.16). Si el número de conjuntos de la partición así obtenida es mayor que 2, ya tenemos la partición que necesitábamos. Si hemos obtenido sólo dos subconjuntos, podemos utilizarlos como V_0 y V_K para generar una configuración K-C mediante el algoritmo descrito anteriormente. Si sólo nos queda un subconjunto, consideraremos que el algoritmo ha fracasado intentando hallar una configuración HC a partir de la componente V_r .

Una vez tenemos los conjuntos B_1, B_2, \dots, B_L , buscaremos los restantes $B_{L+1}, B_{L+2}, \dots, B_{K-1}$. Recordemos que ha de existir un árbol generador T en el grafo de configuración de manera que B_1, B_2, \dots, B_L tengan grado 1 en dicho árbol (es decir, sean *hojas* del árbol), $B_{L+1}, B_{L+2}, \dots, B_{K-1}$ tengan grado

Figura 7.16: Conjuntos B_1, \dots, B_L

mayor o igual que 2 y el camino más corto entre un B_i y B_j , $1 \leq i, j \leq L$ en T contenga, como mínimo, 3 aristas.

Para ello, consideramos el grafo resultante de comprimir en un solo nodo cada conjunto B_i , $i = 1, \dots, L$ así como las restantes componentes R-conexas. En este grafo, hallamos un árbol generador de peso máximo mediante el mismo procedimiento que en el algoritmo de separación de las desigualdades K-C sin usar aristas de peso 0. Si no conseguimos encontrar el árbol de esta manera, lo completamos con aristas de peso 0.

Ahora, comprimimos sucesivamente los vértices de grado 1 (exceptuando los vértices B_1, B_2, \dots, B_L) como en el algoritmo para las desigualdades K-C, hasta obtener un grafo con los vértices $B_1, B_2, \dots, B_L, B_{L+1}, \dots, B_{K-1}$ (figura 7.17). Si alguno de los vértices B_1, B_2, \dots, B_L tiene grado mayor que 2 en este árbol, no nos servirá, por lo que el algoritmo habrá fracasado. Si no, para cada vértice B_i $1 \leq i \leq L$ incidente con dos vértices B_j, B_k , $L+1 \leq j, k \leq K-1$, escogemos entre (B_i, B_j) y (B_i, B_k) la arista de mayor peso, sustituyéndola por la arista (B_j, B_k) , con lo que conseguimos que todos los vértices B_1, \dots, B_L tengan grado 1. De esta manera, tendremos ya una configuración Honeycomb (figura 7.18).

Veamos ahora si su desigualdad asociada es violada por la solución. Si $e = (B_i, B_j)$, sea $T(e)$ la suma de los pesos de las aristas del camino en T desde B_i hasta B_j . Así se cumple $F(x^*) = \sum_{e \in T} T(e) - 2 \sum_{1 < i < j < L} w(B_i : B_j)$. Comprobamos si la desigualdad Honeycomb es violada por x^* . De no ser así, probaremos a comprimir todos los pares de conjuntos B_i, B_j , $L+1 \leq i, j \leq K-1$ con $T((B_i : B_j)) > 2$ y los pares B_i, B_j $1 \leq i \leq L$, $L+1 \leq j \leq K-1$ con $T((B_i : B_j)) > 2$ si el grado de B_j en T es 2.

En la figura 7.18 podemos observar que la desigualdad Honeycomb $F(x) \geq 14$ obtenida en el ejemplo está violada por la solución fraccionaria.

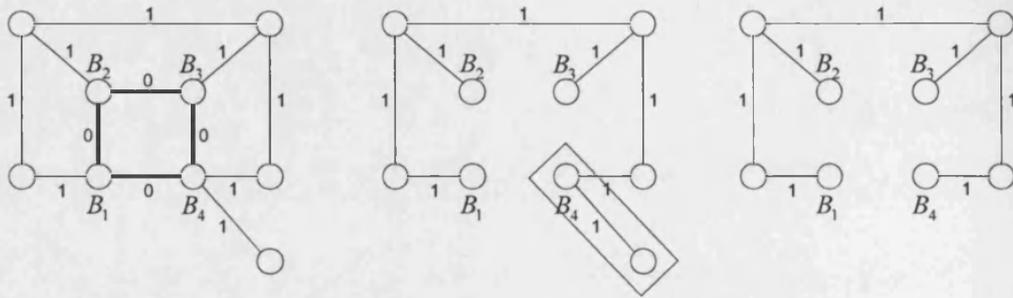


Figura 7.17: Árbol generador y compresión de vértices con grado 1

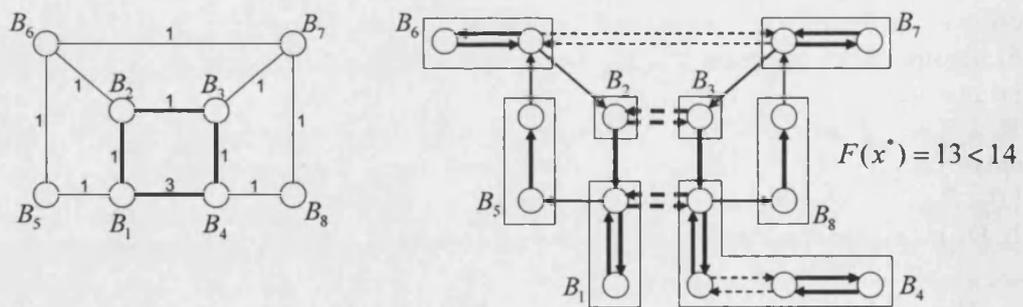


Figura 7.18: Configuración HC con $L=4$, $K=9$, que da lugar a una desigualdad violada por x^*

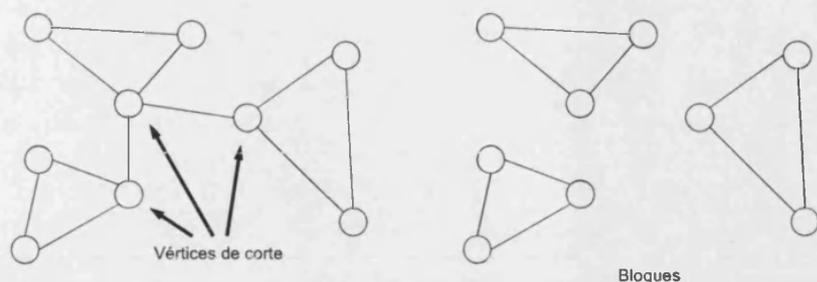


Figura 7.19: Descomposición en bloques

7.8.5. Desigualdades $K-C_{02}$ y $Honeycomb_{02}$

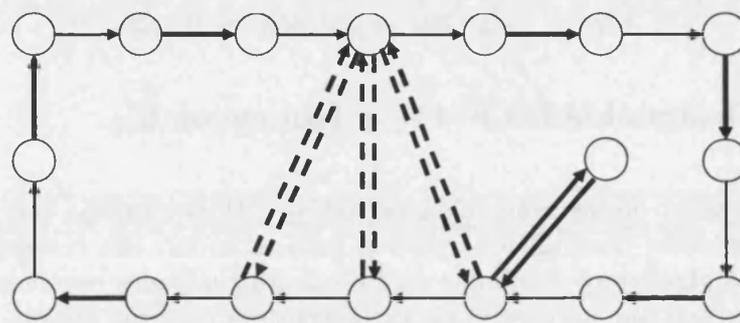
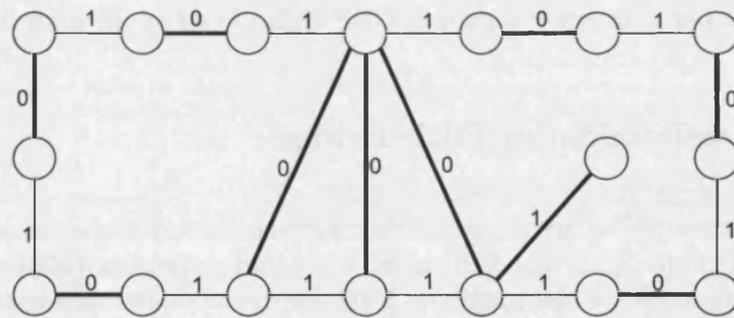
La separación de las desigualdades $K-C_{02}$ y $Honeycomb_{02}$ está basada en los procedimientos desarrollados para el caso estándar. Sin embargo, algunas etapas de los algoritmos diseñadas para estas desigualdades requieren un tratamiento específico que aproveche las características de estas nuevas familias de desigualdades. Puesto que la descripción de los procedimientos heurísticos utilizados es parecida a la presentada para los casos estándar, aunque necesita de un buen número de detalles nuevos, en aras de la brevedad, remitimos al lector al trabajo de Corberán, Mejía y Sanchis (2005bb) donde estas desigualdades son introducidas por primera vez para el GRP definido sobre un grafo mixto.

7.8.6. Desigualdades Path-Bridge

El algoritmo que vamos a describir a continuación fue inicialmente propuesto para el GRP no dirigido por Corberán, Letchford y Sanchis (2001). Este algoritmo de separación de desigualdades Path-Bridge trabaja inicialmente con una pequeña parte del grafo G^* , que llamaremos *bloque*, y, una vez detectada una desigualdad Path-Bridge en dicho subgrafo, trata de extenderla al grafo completo. Un *vértice de corte* será aquél cuya eliminación hace que el grafo resultante pase a ser no conexo. De esta manera, un bloque será un subgrafo conexo maximal de G que no contenga vértices de corte (figura 7.19). Consideraremos la descomposición en bloques del grafo resultante de eliminar de G^* las aristas con peso 0. Escogeremos, pues, uno de estos bloques.

En la figura 7.20 se muestran una solución que viola una desigualdad Path-Bridge y su correspondiente grafo G^* . Sean S_1, S_2, \dots, S_r las componentes R-conexas del bloque elegido. Dadas dos componentes S_i, S_j conectadas por al menos una arista, definimos $\theta(S_i, S_j) = w(\delta(S_i \cup S_j)) + w(S_i : S_j) - 3$. Es fácil ver que si x^* no viola ninguna restricción de conectividad ese valor será siempre mayor o igual que 0. Fijados los parámetros $0 \leq \epsilon_1, \epsilon_2 \leq 1$, para cada par de



Solución no factible x^* Grafo G^* Figura 7.20: Solución x^* que viola una desigualdad PB y grafo G^*

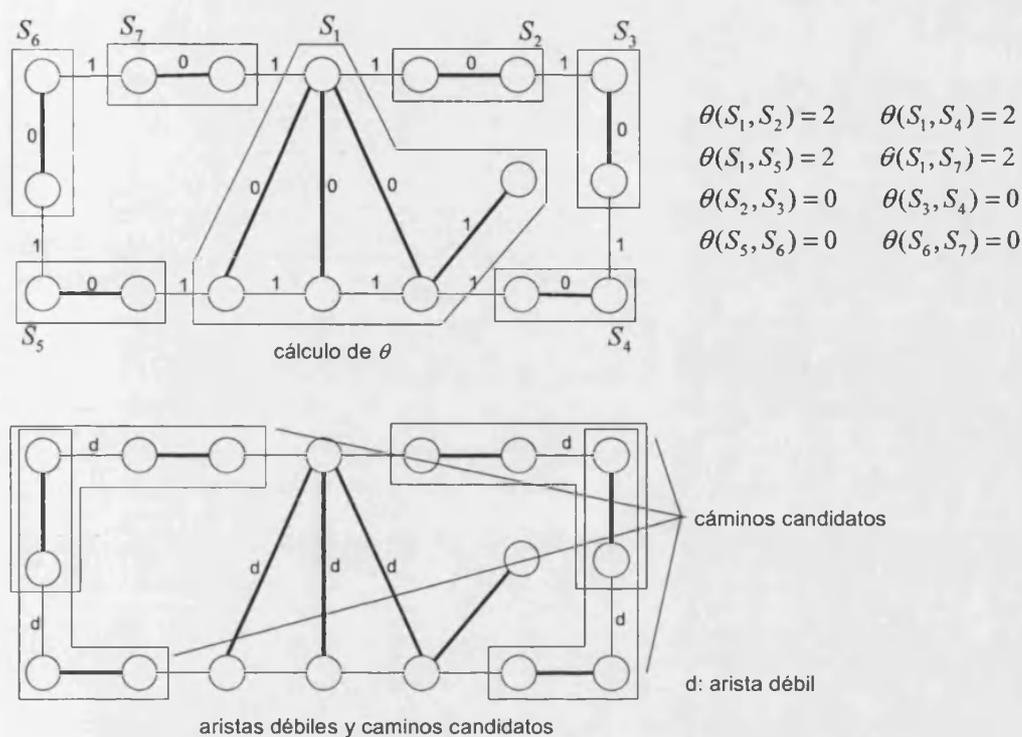


Figura 7.21: Elección de caminos candidatos y etiquetado de aristas débiles

componentes S_j, S_k tales que $\theta(S_j, S_k) \leq \epsilon_1$ etiquetamos las aristas de $(S_j : S_k)$ como *débiles*. Llamaremos en este caso *camino candidato* a (S_j, S_k) y *extremos* del camino a S_j y S_k . Del resto de aristas no etiquetadas, etiquetaremos una arista e como débil si $w_e \leq \epsilon_2$ (ver figura 7.21, donde hemos elegido $\epsilon_1 = 1$ y $\epsilon_2 = 0,5$).

Ahora, eliminamos del grafo G^* las aristas que hemos etiquetado como débiles. Cada una de las componentes conexas resultantes será candidata para dar lugar a los vértices iniciales de los caminos. Así, seleccionada una de estas componentes, que denotaremos por C , definiremos $b = |\delta_R(C)|$ y p como el número de caminos candidatos con un extremo en C y otro fuera. Si $p+b \geq 3$ e impar, $p \geq 1$ y los p caminos candidatos son disjuntos respecto a sus vértices, aceptaremos C como candidata, si no escogeremos otra componente conexa (figura 7.22).

Para cada posible candidata C aceptada, probamos si la desigualdad 2-PB simple asociada a la configuración definida por los p caminos es violada por x^* (figura 7.23). Lo sea o no, continuaremos con el algoritmo para detectar otras posibles desigualdades Path-Bridge violadas.

Escogeremos como $V_1^1, V_1^2, \dots, V_1^p$ los extremos de los caminos en C , como

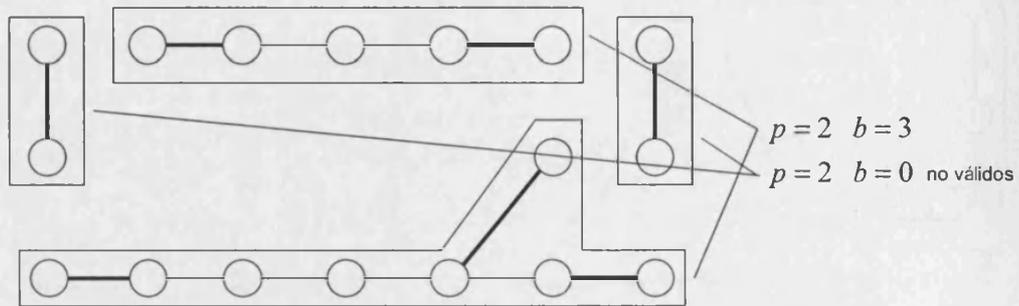


Figura 7.22: Candidatos para inicio de caminos

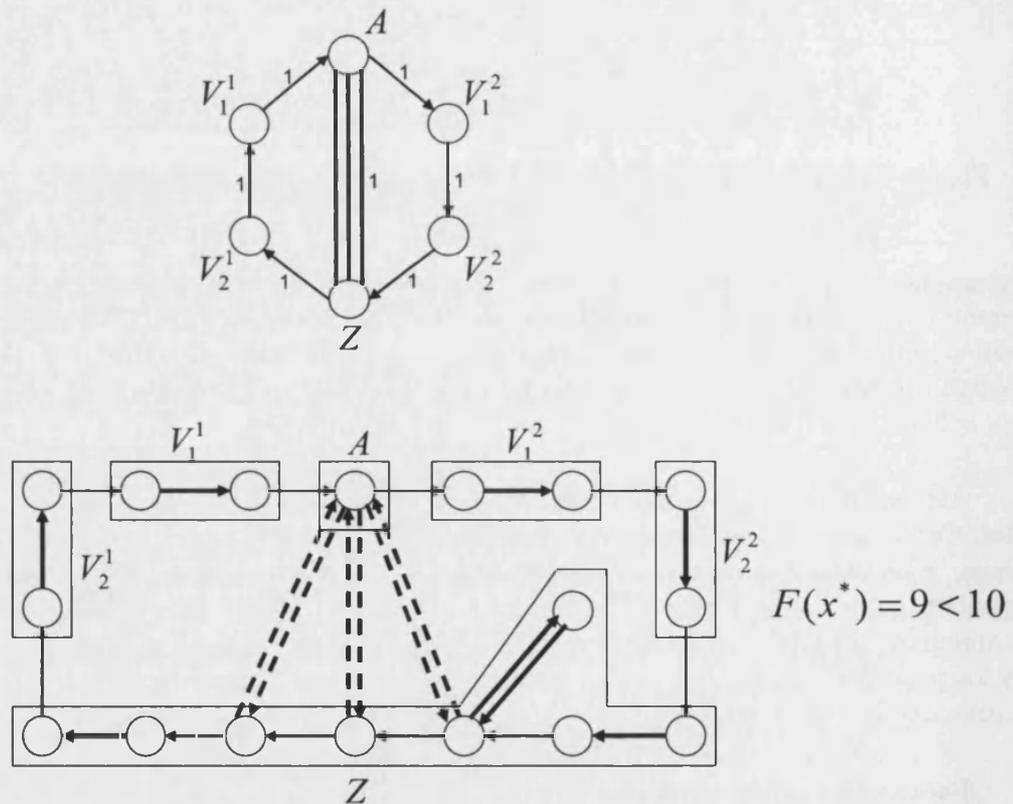
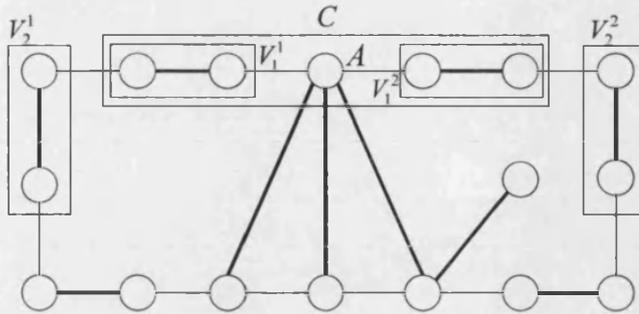
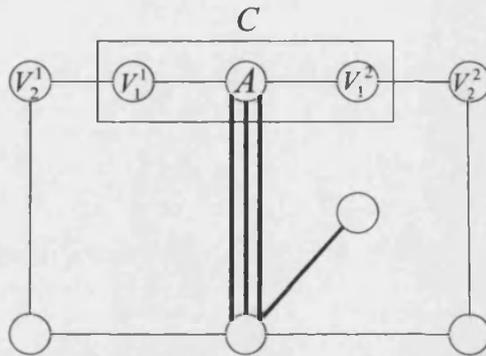


Figura 7.23: Configuración 2-PB asociada a una de las componentes candidatas

Figura 7.24: Elección de A , V_i^1 y V_i^2 Figura 7.25: Grafo comprimido G_s

$V_2^1, V_2^2, \dots, V_2^p$ los extremos fuera de C , y $A = C \setminus \bigcup_{i=1}^p V_1^i$ (figura 7.24). Ahora comprimimos las componentes R-conexas que estén completamente contenidas en C o en $V \setminus C$. Todos los vértices de C incidentes con una arista requerida cuyo extremo opuesto esté en $V \setminus C$ se comprimen en uno solo. Así mismo, comprimimos los vértices de $V \setminus C$ incidentes con alguna arista requerida que tenga el extremo opuesto en C . Al grafo resultante lo llamaremos G_s (figura 7.25). A continuación, procedemos de manera distinta según el valor de b .

Si $b \geq 1$, el vértice comprimido fuera de C incidente con aristas requeridas define el conjunto Z , que será no vacío. Si $x^*((A : Z)) \geq |(A : Z)_R| + 1$, esta estructura no nos servirá para detectar una desigualdad PB violada. En otro caso, continuamos seleccionando aristas como si fuéramos a construir un árbol en G_s , de manera que :

- Todos los vértices de G_s excepto A y V_1^1 son incidentes con alguna arista seleccionada.
- Formen $p + 1$ subárboles disjuntos respecto de las aristas.

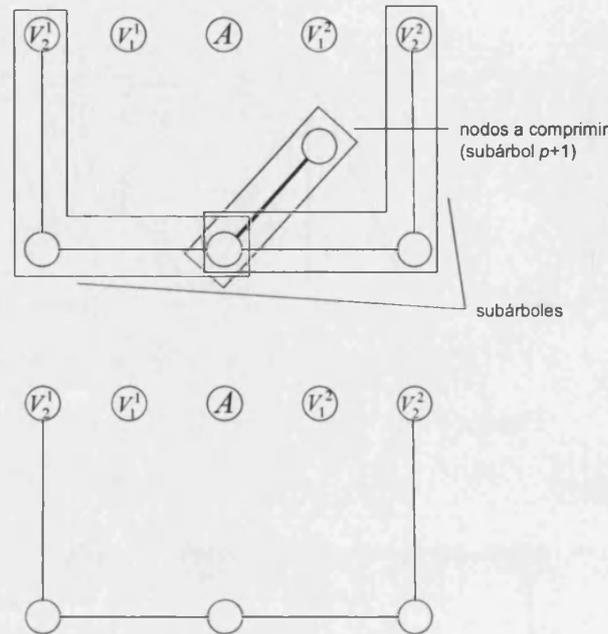


Figura 7.26: Árbol en G_s y compresión de vértices

- El subárbol i contenga un camino de V_2^i a Z pero no de V_2^i a otro V_2^j .
- El subárbol $p + 1$ (puede ser vacío) conecta Z a otros vértices, pero no a V_2^i .

En esta estructura que hemos construido (ver figura 7.26), comprimimos los vértices de grado 1 a excepción de Z y los V_2^i . En la figura 7.27, podemos ver que la desigualdad Path-Bridge así obtenida es violada por la solución fraccionaria.

7.8.7. Desigualdades Zigzag

Para identificar desigualdades Zigzag violadas, hemos diseñado un algoritmo heurístico. Este algoritmo parte de la idea de que en las soluciones que violan desigualdades Zigzag pares es posible hallar una cortadura de aristas no requeridas que contenga un zigzag cuyas variables toman valores fraccionarios en dicha solución. Además, asumiremos que el grafo tiene al menos 2 componentes R-conexas y que no existe ninguna restricción de conectividad violada.

Sea x^* una solución fraccionaria (podemos ver un ejemplo en la figura 7.28). Construimos primero el grafo auxiliar G_{aux}^0 inducido por las aristas (i, j) para las que $x_{ij}^* + x_{ji}^* > 0$. Asignaremos capacidad 1 a aquellas aristas (i, j) no requeridas

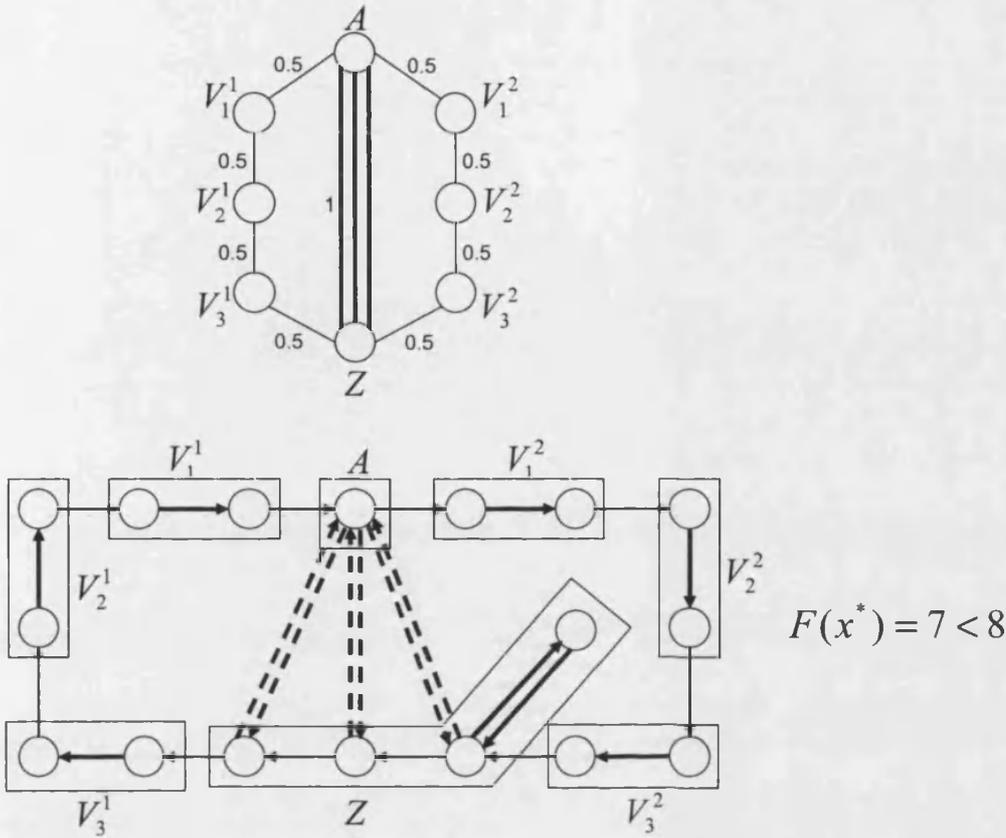


Figura 7.27: Configuración PB regular asociada a una de las componentes candidatas

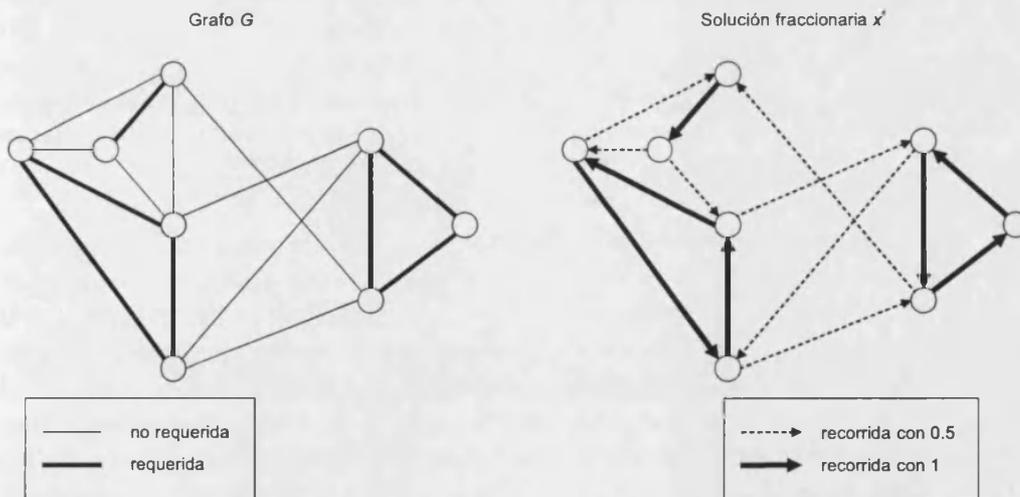


Figura 7.28: Grafo ejemplo y solución fraccionaria para la separación de Z

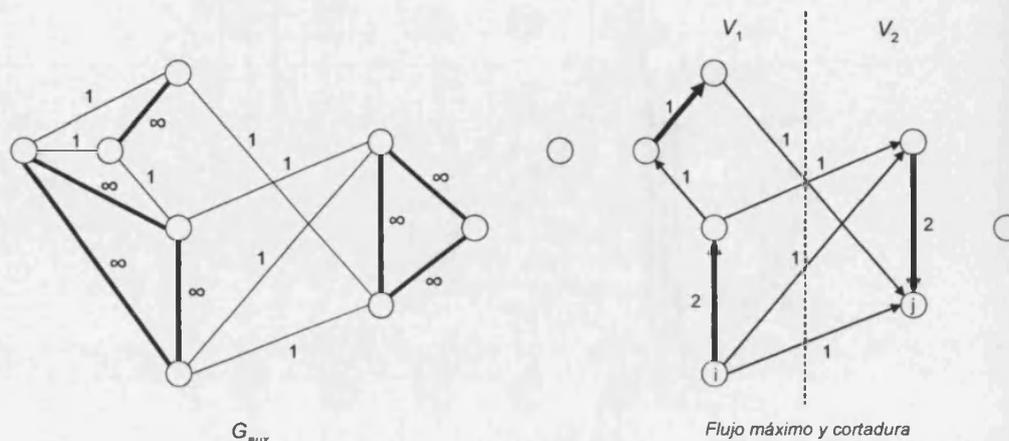


Figura 7.29: Obtención de la cortadura que contenga el zigzag

para las que $x_{ij}^* = 0,5$ y $x_{ji}^* = 0$ (o a la inversa), mientras que el resto de aristas tendrán capacidad infinita.

Sea (i, j) una arista de G_{aux}^0 con capacidad 1. Resolvemos entonces el problema de flujo máximo desde i hasta j . Es fácil ver que el flujo f resultante deberá ser, al menos, 4, ya que, si fuera menor, significaría que existe una restricción de conectividad violada por la solución. Si $f = 4$, identificaremos una cortadura (V_1, V_2) de capacidad 4 (figura 7.29). Esta cortadura contendrá 4 aristas no requeridas con $x_{ij}^* = 0,5$ y $x_{ji}^* = 0$, que serán las que usaremos para intentar definir el zigzag (figura 7.30). Sin embargo, la existencia de dicha cortadura no implica la del zigzag (figura 7.31), por lo que deberemos comprobar esto primero. En el caso de que no existiera, o $f > 4$, no podríamos utilizar la solución del problema de flujo máximo para identificar una desigualdad Zigzag violada. Esto lo hacemos para todas las aristas de G_{aux}^0 con capacidad 1 para obtener el mayor número posible de zigzags, aumentando así la posibilidad de hallar desigualdades violadas.

Supongamos que, efectivamente, hemos hallado un zigzag con dichas aristas. Para identificar mejor los vértices incidentes con las aristas que forman el zigzag, los numeraremos como en la figura 7.32. Nótese que los vértices 1 y 2 (3 y 4, 5 y 6, 7 y 8, respectivamente) pueden ser el mismo, como ocurre en el ejemplo de la figura 7.30, donde coinciden el 3 y el 4, el 5 y el 6, y el 7 y el 8. Procederemos ahora a dividir los conjuntos V_1 y V_2 en dos partes cada uno. Para ello, construimos un nuevo grafo $G_{aux}^1 = (V_1, E_1)$, donde E_1 contendrá las aristas no requeridas entre vértices de V_1 recorridas por la solución y las requeridas tales que $x_{ij}^* + x_{ji}^* > 1$. Estas son las aristas que no deben aparecer en la cortadura que parta V_1 para que la desigualdad Zigzag esté violada por x^* . Si existe una componente conexa en G_{aux}^1 que contenga alguna de las parejas de

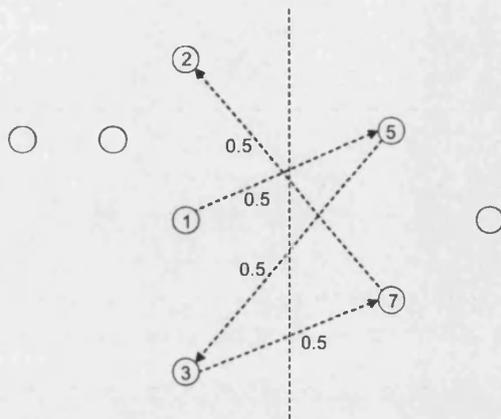


Figura 7.30: Cortadura con zigzag

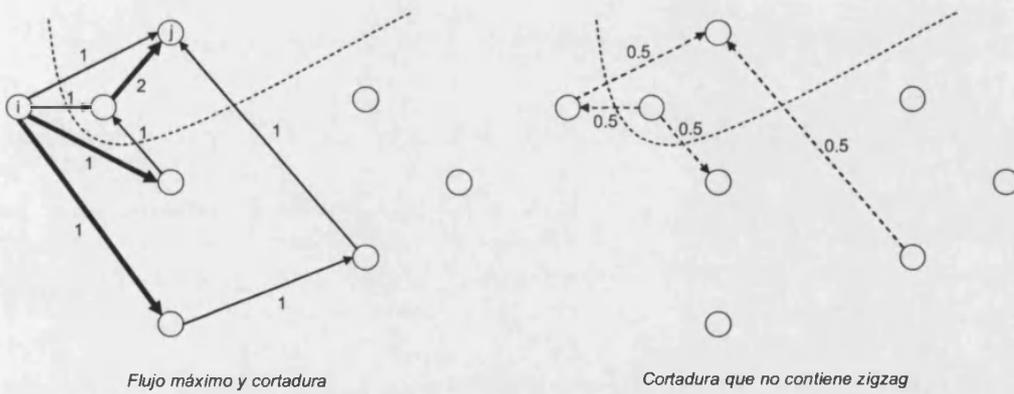


Figura 7.31: Cortadura sin zigzag

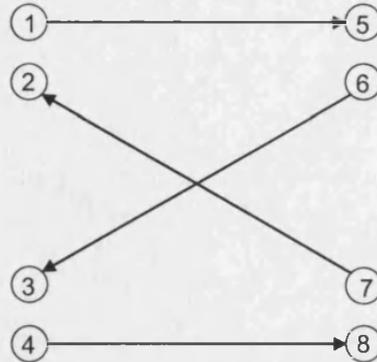
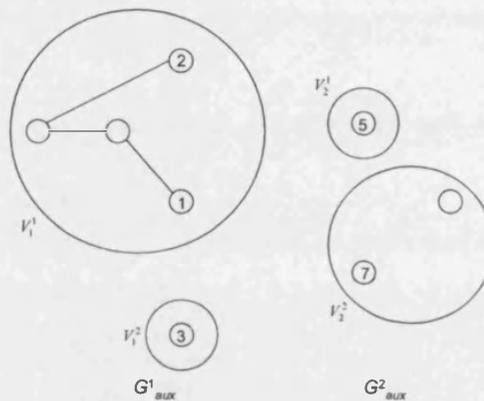


Figura 7.32: Numeración de los vértices del zigzag

Figura 7.33: Partición de los conjuntos V_1 y V_2

vértices $\{1, 3\}$, $\{1, 4\}$, $\{2, 3\}$ o $\{2, 4\}$, no podremos hallar una partición adecuada. Supondremos que esto no ocurre, y consideraremos entonces la partición de V_1 en dos subconjuntos V_1^1 y V_1^2 , donde V_1^1 está formado únicamente por las componentes conexas de G_{aux}^1 que contienen a los vértices 1 y 2, y $V_1^2 = V_1 \setminus V_1^1$ (figura 7.33). De la misma forma podemos crear otra partición considerando las componentes conexas que contienen a los vértices 3 y 4.

Si las dos particiones que hemos hallado para V_1 son R-pares (o ambas son R-impares), tendremos en cuenta una tercera partición, en la que V_1^1 estará formado por las componentes conexas de los vértices 1 y 2 y otra componente conexa R-impar. De esta manera tendremos los dos tipos de particiones posibles, R-pares y R-impares.

Podemos hacer lo mismo para el conjunto V_2 , obteniendo 2 o 3 particiones. Emparejando particiones de V_1 y V_2 de la misma paridad, obtenemos entre

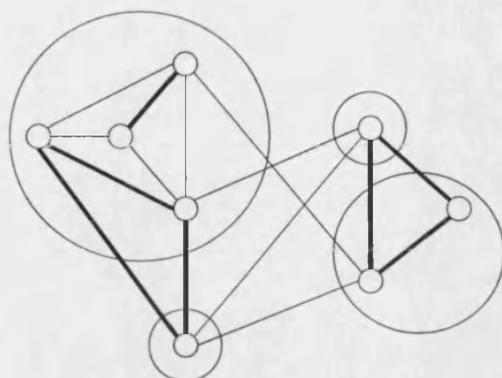


Figura 7.34: Partición final del grafo

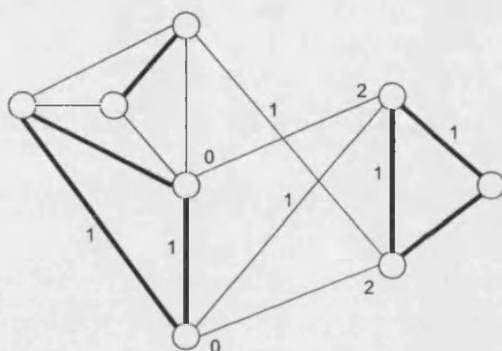


Figura 7.35: Coeficientes de la desigualdad Zigzag identificada

2 y 5 particiones del grafo original en 4 subconjuntos (figura 7.34), a partir de las cuales obtenemos sendas desigualdades Zigzag violadas por la solución fraccionaria (figura 7.35).

En la construcción del grafo G_{aux} para la identificación del zigzag, podemos relajar la condición para que una arista no requerida tenga capacidad 1 a $x_{ij}^* \leq 0,75$ y $x_{ji}^* = 0$, y las desigualdades que hallemos seguirán estando violadas por la solución x^* .

Capítulo 8

Resultados computacionales

8.1. Instancias

Para probar la eficiencia del algoritmo Branch & Cut, hemos utilizado varios conjuntos de instancias de diferentes tamaños y características. Algunos de estos conjuntos ya han sido utilizados con anterioridad en la literatura en el contexto de otros problemas de rutas por arcos, y han sido transformados en instancias del WGRP para este estudio. Otros conjuntos han sido generados de manera aleatoria. A continuación se detalla el proceso seguido para adaptar las instancias ya existentes y generar las nuevas, así como las características de cada uno de estos conjuntos.

8.1.1. Instancias en grafos windy

Instancias del WRPP de Christofides et al.

Este conjunto es el mismo utilizado en Benavent et al. (2003, 2005) para el WRPP y proviene de las 24 instancias propuestas en Christofides et al. (1981) para el RPP. A partir de cada una de estas instancias para el RPP se generan 6 instancias con el mismo grafo pero con diferentes costes asociados a las aristas. Si c_{ij} es el coste asociado a la arista (i, j) en la instancia original, los nuevos costes c_{ij} , c_{ji} se generan siguiendo dos estrategias propuestas por Win (1987):

- **Estrategia 1:** Para cada arista (i, j) , escogemos aleatoriamente dos valores enteros $k_1, k_2 \in [-a, a]$. Los nuevos costes se definen como $c_{ij} =$

$\max\{1, c'_{ij} + k_1\}$, $c_{ji} = \max\{1, c'_{ij} + k_2\}$. Utilizando para a los valores 5, 8 y 10 se obtienen 3 nuevas instancias a partir de cada instancia original.

- **Estrategia 2:** Para cada arista (i, j) , escogemos aleatoriamente dos valores enteros $k_1, k_2 \in [a, b]$. Los nuevos costes se definen como $c_{ij} = k_1$, $c_{ji} = k_2$. Utilizando para $[a, b]$ los valores $[1, 100]$, $[1, 200]$ y $[1, 500]$ se obtienen 3 nuevas instancias a partir de cada instancia original.

Tenemos, por tanto, 144 instancias del WRPP.

Instancias del WRPP de Hertz et al.

Este conjunto de instancias ha sido utilizado en Benavent et al. (2005) y fue generado a partir de las instancias propuestas por Hertz, Laporte y Nanchen (1999) para el RPP. De todas las instancias presentadas en este último trabajo, se han seleccionado las 18 de mayor tamaño. De ellas, 9 corresponden a grafos con una estructura *grid* (rejilla o red), y las otras 9 a grafos cuyos vértices tienen, en su mayoría, grado 4 (conjuntos G y D , respectivamente). Los costes originales son 1 en todos los casos. A partir de cada una de estas instancias, se han generado 6 instancias del WRPP utilizando las mismas estrategias y parámetros que en el conjunto anterior.

Instancias del WRPP de Albaida y Madrigueras

Estas instancias corresponden al WRPP, y han sido utilizadas previamente en Benavent et al. (2003, 2005). Sus grafos asociados representan la red de calles de dos poblaciones españolas, Albaida (Valencia) y Madrigueras (Albacete). Cada arista de estos grafos se escoge como requerida con una cierta probabilidad p . Si queda algún vértice que no sea incidente con ninguna arista requerida, se seleccionan nuevas aristas requeridas entre las que no lo son, marcando de nuevo una arista requerida con probabilidad p . Este proceso se repite hasta que todos los vértices son incidentes con alguna arista requerida. Utilizando 3 valores distintos para p , 0,3, 0,5 y 0,7, y generando dos instancias para cada uno de estos valores de p , se obtienen 6 instancias del RPP para cada uno de los dos grafos. Para generar las instancias del WRPP se emplean las mismas estrategias y los mismos parámetros que en los conjuntos de los dos apartados anteriores. De esta manera, generando dos instancias para cada combinación de parámetros, se obtiene un total de 72 instancias para cada uno de los dos grafos iniciales, agrupadas en los conjuntos Albaida y Madrigueras, respectivamente.

Instancias del WRPP generadas aleatoriamente

Los dos primeros conjuntos de instancias generadas aleatoriamente son los presentados en Benavent et al. (2005). Para generarlas, se seleccionan aleatoriamente $|V|$ puntos en una rejilla 1000×1000 , y se crean aleatoriamente $|E|$ aristas entre estos puntos. A cada arista entre dos vértices i, j se le asigna el coste inicial $c_{ij} = \lfloor b_{ij} + 0,5 \rfloor$, donde b_{ij} es la distancia euclídea entre los puntos i y j . Esta función de costes es la propuesta en la TSPLIB (1991). Si el grafo obtenido no es conexo, se añaden las aristas pertenecientes a 5 árboles generadores distintos. Cada una de las aristas creadas se selecciona como requerida con una probabilidad p . En total, se generan las 18 instancias del RPP correspondientes a las posibles combinaciones de los parámetros $|V| \in \{500, 1000\}$, $|E| \in \{1,5|V|, 2|V|, 3|V|\}$, $p \in \{0,25, 0,50, 0,75\}$. A partir de cada una de ellas, se generan 3 instancias del WRPP utilizando la estrategia 1 mencionada anteriormente con los posibles valores del parámetro $a \in \{50, 80, 100\}$ para la elección de los costes, obteniendo un total de 54 instancias. Estas instancias las agrupamos en dos conjuntos, que denotaremos A500 y A1000, en función del parámetro $|V|$ (número de vértices) a partir del cual han sido generadas.

También se han generado 3 conjuntos de nuevas instancias, intentando imitar las estructuras encontradas en los grafos correspondientes a redes reales, como los de Albaida y Madrigueras. Para generarlos, se han escogido de nuevo $|V|$ puntos en una rejilla 1000×1000 . Por cada vértice se crean g aristas que lo conectan con sus nodos más cercanos respecto de la distancia euclídea, para evitar aristas excesivamente largas que “cruzan” de parte a parte el grafo. Al igual que antes, si el grafo resultante no es conexo, se añaden las aristas de 5 árboles generadores diferentes. Los costes de dichas aristas se han elegido a partir de la distancia euclídea entre los puntos, al igual que en los conjuntos A500 y A1000. Si, dada una arista (i, j) , existe un vértice k tal que $c_{ij} \geq 0,98(c_{ik} + c_{kj})$, se elimina dicha arista para evitar aristas “casi paralelas”. Las aristas requeridas se seleccionan con probabilidad p , como en casos anteriores. Utilizando, al igual que en casos anteriores, la estrategia 1 de Win, se modifican los costes de las aristas para que no sean simétricos. Para las posibles combinaciones de valores de los parámetros $|V| \in \{500, 750, 1000\}$, $p \in \{0,25, 0,50, 0,75\}$, $g \in [3, 6]$ y $a \in \{10, 20\}$ se han generado 72 instancias, divididas en 3 conjuntos B500, C750 y D1000 según el número de vértices a partir del cual se han generado los correspondientes grafos, con 24 instancias cada uno.

El grafo en todas estas instancias ha sido simplificado para que todos los vértices sean incidentes con aristas requeridas, por lo que el número final de vértices puede ser menor que $|V|$.

Instancias del WGRP generadas aleatoriamente

Sin simplificar los 18 grafos generados aleatoriamente que han dado lugar a las instancias de los conjuntos A500 y A1000 y considerando todos los vértices como requeridos, se han creado otras 54 instancias escogiendo los costes de manera análoga a la de los conjuntos A500 y A1000. Así obtenemos dos conjuntos con 27 instancias del WGRP cada uno, de 500 y 1000 vértices respectivamente, llamados GA500 y GA1000.

Hemos generado otro conjunto, llamado GB500, que consta de 24 instancias generadas aleatoriamente siguiendo el mismo proceso que para las instancias del conjunto B500, con la diferencia de que, en este caso, el grafo no se ha simplificado, permitiendo así la existencia de vértices no incidentes con aristas requeridas, que serán los vértices requeridos.

8.1.2. Instancias en grafos mixtos

Estas instancias fueron generadas por Corberán, Mota y Sanchis (2005bb), formando un total de 153.

Instancias del MGRP de Albaida, Aldaya y Madrigueras

Estas 81 instancias fueron generadas por Corberán, Mejía y Sanchis (2005aa) a partir de los grafos de Albaida y Madrigueras mencionados anteriormente, más el correspondiente a una tercera ciudad española, Aldaya (Valencia). Cada arista es transformada en arco con probabilidad $p \in \{0,1,0,3,0,5,0,7,0,9\}$ y marcada como requerida con probabilidad $q \in \{0,1,0,3,0,5,0,7,0,9\}$. Si el grafo resultante no es fuertemente conexo, algunos de los arcos son convertidos en aristas, hasta que sólo queda una componente fuertemente conexa. Para los grafos de Albaida y Madrigueras se generó una instancia con cada combinación posible de los parámetros p y q , dando lugar a un total de 50 instancias. Para el grafo de Aldaya, se generaron 25 instancias de manera análoga a las anteriores, más otras 5 usando los parámetros $p = q \in \{0,0,2,0,4,0,6,0,8\}$ (obsérvese que la instancia con $p = q = 0$ es, en realidad, una instancia del GTSP). Otra instancia más asociada al grafo de Aldaya representa un problema real de recogida de basuras resuelto en dicha ciudad. En total, hay 31 instancias generadas a partir del grafo de Aldaya, sumando un total de 81 instancias del MGRP. Éstas han sido divididas en tres conjuntos, M-Alba, M-Alda y M-Madr, uno por ciudad.

Instancias del MGRP generadas aleatoriamente

Estas instancias, también generadas por Corberán et al. (2005bb), fueron creadas usando el mismo procedimiento que las instancias B500, C750 y D1000 para el WRPP, sólo que ahora, el número de árboles generadores utilizados para conectar el gafo en el caso en que éste no fuera conexo es igual al número de aristas creadas para cada vértice, que viene dado por el parámetro d . Así, se generaron 36 instancias utilizando las posibles combinaciones de los siguientes parámetros: número de vértices $|V| \in \{500, 1000\}$, grado (y número de árboles generadores añadidos) $d \in \{4, 5\}$, probabilidad de que una arista sea transformada en arco $p \in \{0,25, 0,5, 0,75\}$ y probabilidad de que un enlace sea requerido $q \in \{0,25, 0,5, 0,75\}$. Hemos dividido estas instancias en dos conjuntos, MGB500 y MGD1000, según el número de vértices de los grafos.

Instancias del MRPP generadas aleatoriamente

El método seguido por Corberán et al. (2005bb) para generar estas instancias es el mismo que para las instancias anteriores del MGRP. Para que las instancias resultantes no contuvieran vértices requeridos, se simplificó el grafo al igual que en las instancias para el WRPP. El número de instancias así generadas es 36, divididas en dos conjuntos llamados MRB500 y MRD1000 de acuerdo al número de vértices de los grafos antes de la simplificación.

8.1.3. Instancias en grafos no dirigidos

Estas instancias fueron generadas por Corberán, Letchford y Sanchis (2001). Suman un total de 47, 40 de ellas pertenecientes al GRP y otras 7 al GTSP.

Instancias del GRP de Albaida y Madrigueras

Para generar estas instancias se utilizaron los grafos correspondientes a las ciudades de Albaida y Madrigueras. Las aristas requeridas fueron seleccionadas *visualmente* (por ejemplo, seleccionando las aristas orientadas de norte a sur, de este a oeste, formando ciclos, ...). Así se obtuvieron un total de 10 instancias, incluidas en el conjunto U-GRP.

Instancias del GRP generadas aleatoriamente

A partir de los grafos de la ciudad de Albaida y Madrigueras, se seleccionaron con probabilidad $p \in \{0,3,0,5,0,7\}$ las aristas requeridas. Generando 5 instancias para cada grafo y cada posible valor de p , se obtuvieron 30 instancias, que aquí reciben el nombre de U-Alba y U-Madr, según la ciudad que dio lugar al grafo.

Instancias del GTSP

Este conjunto se obtuvo a partir de 7 instancias euclídeas de la TSPLIB con grafos planares. Para hacer los grafos menos densos, se eliminaron todas las aristas excepto aquéllas pertenecientes a un tour óptimo y las que conectaran a un vértice con alguno de los tres vértices más cercanos. Haremos referencia al conjunto de estas 7 instancias como GTSP.

8.1.4. Características de las instancias

En las siguientes tablas pueden observarse las características de cada conjunto de instancias. La tabla 8.1 muestra información acerca del número de vértices de los grafos de cada conjunto (promedio, mínimo y máximo), mientras que la tabla 8.2 muestra el número de vértices requeridos aislados en aquellos conjuntos de instancias de problemas *generales* (en los *rurales* no hay vértices requeridos aislados). Los detalles acerca del número total de aristas y de aristas requeridas aparecen en las tablas 8.3 y 8.4 respectivamente. También, para las instancias definidas en grafos mixtos, se muestra el número total de arcos (tabla 8.5) y de arcos requeridos (tabla 8.6). Por último, podemos ver el número de componentes R-conexas para cada conjunto de instancias en la tabla 8.7.

8.2. Pruebas computacionales

Las primeras pruebas computacionales realizadas mostraron que las instancias de los conjuntos de CHR, HG, HD, ALB y MAD eran resueltas fácilmente por el algoritmo Branch & Cut en cuestión de segundos. Por lo tanto, para las siguientes pruebas, hemos decidido utilizar los conjuntos de instancias A500, A1000, B500, C750 y D1000. El ordenador empleado para todas las pruebas ha sido un Pentium IV a 1.7GHz, con 512MB de RAM. El algoritmo está programado en C/C++, utilizando el MIP Solver de Cplex 9.0, con Concert Technology 2.0.

| Conjunto | Problema | Promedio | Mín. | Máx. |
|-------------|----------|----------|------|------|
| CHR | WRPP | 25.125 | 7 | 50 |
| HERTZ G | WRPP | 83.444 | 60 | 100 |
| HERTZ D | WRPP | 84.556 | 68 | 100 |
| Albaida | WRPP | 116 | 116 | 116 |
| Madrigueras | WRPP | 196 | 196 | 196 |
| A500 | WRPP | 400.778 | 265 | 488 |
| A1000 | WRPP | 848.333 | 599 | 988 |
| B500 | WRPP | 446.083 | 318 | 498 |
| C750 | WRPP | 673.417 | 502 | 750 |
| D1000 | WRPP | 895.083 | 661 | 999 |
| GA500 | WGRP | 500 | 500 | 500 |
| GB500 | WGRP | 500 | 500 | 500 |
| M-Alba | MGRP | 116 | 116 | 116 |
| M-Madr | MGRP | 196 | 196 | 196 |
| M-Alda | MGRP | 214 | 214 | 214 |
| MGB500 | MGRP | 500 | 500 | 500 |
| MGD1000 | MGRP | 1000 | 1000 | 1000 |
| MRB500 | MRPP | 449 | 357 | 498 |
| MRD1000 | MRPP | 899.667 | 708 | 999 |
| U-Alba | GRP | 116 | 116 | 116 |
| U-Madr | GRP | 196 | 196 | 196 |
| U-GRP | GRP | 116 | 116 | 116 |
| GTSP | GTSP | 181.714 | 150 | 225 |

Tabla 8.1: Tipo de problema y número de vértices de cada conjunto de instancias

| Conjunto | Promedio | Mín. | Máx. |
|----------|----------|------|------|
| GA500 | 73.444 | 6 | 235 |
| GB500 | 57.250 | 0 | 190 |
| M-Alba | 30.840 | 0 | 93 |
| M-Madr | 47.200 | 1 | 146 |
| M-Alda | 49.677 | 1 | 214 |
| MGB500 | 51 | 2 | 143 |
| MGD1000 | 100.333 | 1 | 292 |
| U-Alba | 24.200 | 6 | 57 |
| U-Madr | 30.600 | 4 | 69 |
| U-GRP | 24.667 | 6 | 38 |
| GTSP | 181.714 | 150 | 225 |

Tabla 8.2: Número de vértices requeridos aislados de las instancias

| Conjunto | Promedio | Mín. | Máx. |
|----------|----------|------|------|
| CHR | 58.917 | 10 | 184 |
| HG | 149.333 | 105 | 180 |
| HD | 172.889 | 141 | 193 |
| ALB | 174 | 174 | 174 |
| MAD | 316 | 316 | 316 |
| A500 | 1268.222 | 842 | 1719 |
| A1000 | 2521.778 | 1656 | 3952 |
| B500 | 1131.5 | 630 | 1537 |
| C750 | 1706 | 1013 | 2288 |
| D1000 | 2286.667 | 1297 | 3073 |
| GA500 | 1135 | 855 | 1505 |
| GB500 | 1209.833 | 887 | 1549 |
| M-Alba | 113.105 | 84 | 164 |
| M-Madr | 192.480 | 118 | 298 |
| M-Alda | 224 | 224 | 224 |
| MGB500 | 620.722 | 311 | 1021 |
| MGD1000 | 1239.889 | 611 | 2033 |
| MRB500 | 583 | 261 | 987 |
| MRD1000 | 1178.056 | 521 | 1969 |
| U-Alba | 174 | 174 | 174 |
| U-Madr | 316 | 316 | 316 |
| U-GRP | 174 | 174 | 174 |
| GTSP | 329.857 | 296 | 392 |

Tabla 8.3: Número total de aristas de las instancias

| Conjunto | Promedio | Mín. | Máx. |
|----------|----------|------|------|
| CHR | 28.083 | 4 | 78 |
| HG | 77.222 | 41 | 113 |
| HD | 85.222 | 50 | 121 |
| ALB | 96.333 | 83 | 129 |
| MAD | 172.889 | 152 | 230 |
| A500 | 481.333 | 190 | 770 |
| A1000 | 1148.556 | 450 | 2229 |
| B500 | 610 | 222 | 1179 |
| C750 | 918.333 | 362 | 1669 |
| D1000 | 1221.583 | 469 | 2352 |
| GA500 | 571.667 | 190 | 1135 |
| GB500 | 599 | 225 | 1158 |
| M-Alba | 71.368 | 26 | 148 |
| M-Madr | 96.360 | 13 | 250 |
| M-Alda | 106.935 | 0 | 209 |
| MGB500 | 319.778 | 92 | 759 |
| MGD1000 | 627.222 | 171 | 1515 |
| MRB500 | 316.889 | 86 | 739 |
| MRD1000 | 629 | 159 | 1481 |
| U-Alba | 85.733 | 44 | 122 |
| U-Madr | 158.267 | 86 | 238 |
| U-GRP | 76.889 | 52 | 126 |
| GTSP | 0 | 0 | 0 |

Tabla 8.4: Número de aristas requeridas de las instancias

| Conjunto | Promedio | Mín. | Máx. |
|----------|----------|------|------|
| M-Alba | 60.895 | 10 | 90 |
| M-Madr | 123.520 | 18 | 198 |
| M-Alda | 127 | 127 | 127 |
| MGB500 | 597.278 | 277 | 991 |
| MGD1000 | 1209.944 | 532 | 2031 |
| MRB500 | 550.833 | 210 | 976 |
| MRD1000 | 1137.278 | 470 | 1984 |

Tabla 8.5: Número total de arcos de las instancias



| Conjunto | Promedio | Mín. | Máx. |
|----------|----------|------|------|
| M-Alba | 36.474 | 9 | 74 |
| M-Madr | 61.360 | 2 | 158 |
| M-Alda | 60.613 | 0 | 118 |
| MGB500 | 289.889 | 62 | 729 |
| MGD1000 | 602.944 | 136 | 1545 |
| MRB500 | 292.778 | 69 | 676 |
| MRD1000 | 601.167 | 131 | 1502 |

Tabla 8.6: Número de arcos requeridos de las instancias

| Conjunto | Promedio | Mín. | Máx. |
|----------|----------|------|------|
| CHR | 4.042 | 1 | 7 |
| HG | 13.333 | 4 | 20 |
| HD | 4.042 | 1 | 7 |
| ALB | 14.111 | 9 | 22 |
| MAD | 23 | 7 | 33 |
| A500 | 33.889 | 1 | 76 |
| A1000 | 49.333 | 1 | 150 |
| B500 | 37.917 | 1 | 102 |
| C750 | 55.833 | 1 | 152 |
| D1000 | 76.083 | 1 | 202 |
| GA500 | 99.222 | 7 | 311 |
| GB500 | 93.333 | 1 | 281 |
| M-Alba | 25.789 | 1 | 70 |
| M-Madr | 65.040 | 2 | 168 |
| M-Alda | 69.129 | 2 | 214 |
| MGB500 | 86.833 | 3 | 245 |
| MGD1000 | 169.167 | 2 | 480 |
| MRB500 | 35.833 | 1 | 102 |
| MRD1000 | 68.833 | 1 | 188 |
| U-Alba | 38.200 | 11 | 73 |
| U-Madr | 53.400 | 7 | 112 |
| U-GRP | 48.222 | 11 | 65 |
| GTSP | 181.714 | 150 | 225 |

Tabla 8.7: Número de componentes R-conexas de las instancias

Inicialmente, la fase de Planos de Corte del algoritmo Branch & Cut sigue un esquema similar al propuesto por Benavent et al. (2003), aunque añadiendo nuevos algoritmos de separación para conjuntos de desigualdades que definen faceta del WGRP. A continuación, aparece detallado el esquema del algoritmo de planos de corte, donde $ncon$, $nimp$ y nkc son el número de desigualdades de conectividad, R-imparidad y K-C violadas halladas en cada iteración, respectivamente, y $ntot$ el número total de desigualdades violadas halladas por el algoritmo en la presente iteración.

1. $ntot = ncon = nimp = nkc = 0$.
2. Hacer:
 - 2.1. Ejecutar algoritmo **heurístico** para desigualdades de **Conectividad** con $\epsilon = 0$.
 - 2.2. Ejecutar algoritmo **heurístico** para desigualdades de **R-imparidad** con $\epsilon = 0$.
 - 2.3. Si $nimp = 0$, ejecutar algoritmo **heurístico** para desigualdades de **R-imparidad** con $\epsilon = 0,25$.
 - 2.4. Si $nimp = 0$, ejecutar algoritmo **heurístico** para desigualdades de **R-imparidad** con $\epsilon = 0,5$.
 - 2.5. Si $ncon = 0$, ejecutar algoritmo **heurístico** para desigualdades de **Conectividad** con $\epsilon = 0,25$.
 - 2.6. Si $ncon = 0$, ejecutar algoritmo **heurístico** para desigualdades de **Conectividad** con $\epsilon = 0,5$.
 - 2.7. Si $ncon = 0$, ejecutar algoritmo **exacto** para desigualdades de **Conectividad**.
 - 2.8. Si $ntot = 0$, ejecutar algoritmo **exacto** para desigualdades de **R-imparidad**.
 - 2.9. Si $ntot < 10$
 - 1) Ejecutar algoritmo **heurístico** para desigualdades **K-C** y **K-C₀₂** con $\epsilon = 0$.
 - 2) Si $nkc = 0$, ejecutar algoritmo **heurístico** para desigualdades **K-C** y **K-C₀₂** con $\epsilon = 0,25$.
 - 3) Si $nkc = 0$, ejecutar algoritmo **heurístico** para desigualdades **K-C** y **K-C₀₂** con $\epsilon = 0,5$.
 - 4) Si $nkc = 0$, ejecutar algoritmo **heurístico** para desigualdades **Zigzag**.
 - 2.10. Si $ntot < 10$, ejecutar algoritmo **heurístico** para desigualdades **Honeycomb** y **Honeycomb₀₂**.
 - 2.11. Si $ntot < 15$, ejecutar algoritmo **heurístico** para desigualdades **Path-Bridge**.

| | Óptimos (total/%) | Tiempo (segs.) | Nodos | Gap |
|-------|-------------------|----------------|-------|---------|
| A500 | 27 / 100 % | 172.2 | 43.8 | 0 % |
| A1000 | 21 / 77.8 % | 8395.8 | 127.1 | 0.022 % |
| B500 | 24 / 100 % | 245.4 | 26.5 | 0 % |
| C750 | 23 / 95.8 % | 4607.5 | 144.9 | 0.027 % |
| D1000 | 18 / 75 % | 10352 | 90.1 | 0.286 % |
| Total | 113 / 89.7 % | 4732.2 | 86.4 | 0.063 % |

Tabla 8.8: Primera prueba computacional

2.12. Si $ntot = 0$, ejecutar algoritmo heurístico para desigualdades K-C uniendo pares de componentes R-conexas.

Mientras $ntot \neq 0$.

En la tabla 8.8, podemos ver un resumen de los resultados obtenidos al ejecutar el algoritmo de ramificación y corte con un tiempo límite de 10 horas para cada instancia. La primera columna muestra el número de instancias en las que el algoritmo alcanzó la solución óptima en el tiempo límite (número total y porcentaje sobre cada conjunto de instancias). La segunda columna muestra el tiempo medio empleado en la resolución de las instancias de cada conjunto. En la tercera columna aparece el número medio de nodos del árbol de búsqueda para cada conjunto, y en la cuarta el *gap* medio.

Tras observar los tiempos empleados por las distintas fases del algoritmo de planos de corte, comprobamos que más del 50 % del tiempo se emplea en el algoritmo heurístico (el más complejo) para detectar desigualdades K-C y K-C₀₂ violadas uniendo componentes R-conexas, a pesar de que únicamente se utiliza en aquella iteración en la que ningún otro algoritmo ha sido capaz de identificar ningún corte. También el algoritmo heurístico para las desigualdades K-C y K-C₀₂ con $\epsilon \neq 0$ consume bastante tiempo, más del 12 %. Por otra parte, aunque el algoritmo de identificación de desigualdades Zigzag no gasta tanto tiempo, aproximadamente un 7 % del total, el número de desigualdades violadas que identifica es bastante reducido.

Por todo ello, decidimos aplicar algunas modificaciones al algoritmo de planos de corte. Para ahorrar tiempo en cada nodo del árbol de búsqueda y así poder analizar más nodos, decidimos ejecutar el heurístico para las desigualdades K-C y K-C₀₂ con $\epsilon = 0,5$ sólo en la fase de planos de corte del nodo raíz. Igualmente hicimos con el algoritmo heurístico que prueba a unir pares de componentes R-conexas para buscar desigualdades K-C y K-C₀₂. Aunque el algoritmo exacto para las desigualdades de R-imparidad encuentra generalmente un gran número de desigualdades violadas, consume también mucho tiempo, por lo que hemos decidido utilizarlo sólo en una de cada dos iteraciones del algoritmo

| | Óptimos (total/%) | Tiempo (seg.) | Nodos | Gap |
|-------|-------------------|---------------|-------|---------|
| A500 | 27 / 100 % | 81 | 43.4 | 0 % |
| A1000 | 23 / 85.2 % | 7402 | 200.5 | 0.010 % |
| B500 | 24 / 100 % | 287.6 | 61.7 | 0 % |
| C750 | 22 / 91.7 % | 5199.2 | 185.4 | 0.033 % |
| D1000 | 18 / 75 % | 10846.3 | 162.3 | 0.267 % |
| Total | 114 / 90.5 % | 4714.6 | 130.3 | 0.059 % |

Tabla 8.9: Segunda prueba computacional

| | Óptimos (total/%) | Tiempo (seg.) | Nodos | Gap |
|-------|-------------------|---------------|-------|---------|
| A500 | 27 / 100 % | 59.4 | 37.4 | 0 % |
| A1000 | 26 / 96.3 % | 2926.2 | 220.2 | 0.187 % |
| B500 | 24 / 100 % | 461.1 | 63.1 | 0 % |
| C750 | 22 / 91.7 % | 6611.6 | 246 | 0.025 % |
| D1000 | 17 / 70.8 % | 11782.7 | 270.2 | 0.542 % |
| Total | 116 / 92.1 % | 4231.3 | 165.5 | 0.108 % |

Tabla 8.10: Tercera prueba computacional

de planos de corte en cada nodo del árbol. Por último, se ha intentado reducir la frecuencia con la que se llama al algoritmo heurístico de identificación de desigualdades Path-Bridge, cambiando el criterio para su llamada a que el número total de cortes encontrados $ntot$ sea menor que 8. También decidimos emplear un algoritmo metaheurístico de Búsqueda Dispersa (Benavent et al., 2005) en las instancias en las que el gap en el nodo raíz fuera mayor del 1%. En la tabla 8.9, podemos observar los resultados obtenidos tras estas modificaciones.

Podemos ver que con estas modificaciones, el algoritmo obtiene un óptimo más, y el gap medio, en general, ha mejorado ligeramente. El tiempo medio empleado es muy similar, pero el número de nodos explorados, para los conjuntos con las instancias más grandes, ha aumentado.

Al analizar las instancias que no se han podido resolver óptimamente, comprobamos que en ellas el gap en el nodo raíz es bastante más grande que en el resto de instancias, y resulta insalvable posteriormente para el algoritmo. La experiencia con instancias de tamaño mediano nos mostró que, en general, la cota inferior en el nodo raíz suele estar más cercana al valor óptimo que la cota superior. Por ello decidimos aplicar una estrategia de reinicio con una cota superior artificial tal y como se describe en el capítulo anterior. Los resultados obtenidos con esta nueva estrategia se pueden observar en la tabla 8.10.

En primer lugar, observamos que el número de instancias resueltas ha au-

mentado, en concreto se resuelven 3 instancias más del conjunto A1000, aunque una del conjunto D1000 que antes se resolvía ahora no llega al óptimo. El tiempo empleado disminuye considerablemente en las instancias A500 y A1000, mientras que aumenta ligeramente en el resto. Este comportamiento puede deberse a que estas instancias ya estaban muy cerca de ser resueltas óptimamente en las pruebas anteriores (con un *gap* medio de sólo el 0.010%). Al utilizar una cota superior, saturamos rápidamente algunos nodos que no van a conducirnos a la solución óptima, con lo que el algoritmo tiene tiempo para explorar algunos nodos más, que son suficientes para conseguir resolver de manera óptima la instancia. Además, en estas instancias, al estar generadas de manera completamente aleatoria y carecer de una estructura clara, aparecen menos desigualdades K-C, Honeycomb, Path-Bridge y Zigzag violadas, por lo que la fase de separación no es de gran ayuda para salvar el *gap* y necesitamos de alguna estrategia que nos permita explorar más nodos del árbol de búsqueda.

En las pruebas realizadas hasta ahora, hemos empleado la estrategia de ramificación por restricciones descrita en el capítulo anterior y, en el caso en que no se encuentren restricciones para ramificar, se escoge la variable con parte fraccionaria más grande. Hemos llevado a cabo una serie de pruebas sobre una muestra reducida con las 23 instancias más difíciles de los conjuntos A1000, C750, D1000 y GB con diferentes estrategias de ramificación. El esquema de separación utilizado en estos experimentos, es el de las últimas pruebas descritas, incluyendo también el uso de la cota superior artificial y los reinicios. Hemos probado un total de cuatro estrategias diferentes:

- Ramificación por restricciones y por la variable con parte fraccionaria más cercana a 0.5 en caso de no hallar restricciones (Rest+variable).
- *Strong branching* (SB).
- Ramificación por restricciones y Strong Branching en caso de no hallar restricciones (Rest+SB).
- Ramificación por restricciones y por variable a elección del Cplex en caso de no hallar restricciones (Rest+Cplex).

Los resultados de estas pruebas pueden observarse en la tabla 8.11. Podemos ver que la estrategia de *strong branching* es la que mejores resultados obtiene.

Finalmente, hemos extendido las pruebas a todos los conjuntos de instancias descritos al comienzo de este capítulo, utilizando la ramificación de tipo *strong branching*. La tabla 8.12 muestra un resumen de los resultados obtenidos en estas pruebas. Para las instancias en grafos mixtos y las del GTSP no hemos empleado los algoritmos heurísticos y metaheurísticos para la obtención de una cota superior, ya que dichos algoritmos no están preparados para funcionar

| | Óptimos (total/ %) | Tiempo (seg.) | Nodos | Gap |
|---------------|--------------------|---------------|--------|---------|
| Rest+variable | 7 / 30.4 % | 30185.8 | 1025.9 | 1.544 % |
| SB | 11 / 47.8 % | 22399.8 | 608.3 | 1.089 % |
| Rest+SB | 7 / 30.4 % | 28917.2 | 740.1 | 1.261 % |
| Rest+Cplex | 7 / 30.4 % | 31481.2 | 1041.7 | 1.718 % |

Tabla 8.11: Prueba con distintas estrategias de ramificación

en este tipo de instancias. Por esta razón, en 3 de las instancias del conjunto MGD1000, el algoritmo de Ramificación y Corte finalizó sin haber hallado ninguna solución factible, es decir, sin cota superior. El *gap* medio presentado para este conjunto no incluye, pues, los resultados de estas 3 instancias. La primera columna muestra el número total de instancias de cada conjunto que han sido resueltas óptimamente en el tiempo límite de 10 horas, así como el porcentaje sobre el total de instancias de cada conjunto. El tiempo medio empleado para la resolución, medido en segundos, se puede observar en la segunda columna, mientras que la tercera muestra el promedio de nodos del árbol de enumeración explorados (sin contar el nodo raíz). Finalmente, podemos ver el *gap* medio entre cota superior e inferior para cada conjunto.

Podemos observar que el algoritmo Branch & Cut es capaz de resolver óptimamente la mayor parte de las instancias. Sólo en 3 instancias del conjunto C750, 4 del D1000, 5 del GB500, 5 del MGD1000 y 1 del GTSP no ha podido terminar óptimamente en el tiempo límite de 10 horas. Todas estas instancias tienen en común el elevado número de componentes R-conexas, la mayoría con más de 200 y hasta 480. Cuando el número de componentes R-conexas es menor de 200, el algoritmo no parece tener excesivas dificultades en alcanzar la solución óptima.

Por otra parte, parece que las instancias sobre grafos windy son más difíciles de resolver que sobre grafos mixtos. Si comparamos los resultados para los conjuntos GB500 y MGB500, que fueron generados de manera análoga y con características similares, podemos ver que para grafos mixtos todas las instancias han sido resueltas, mientras que en 5 casos para grafos windy esto no fue posible.

En la tabla 8.13 podemos observar el promedio de desigualdades violadas de cada tipo halladas por los algoritmos de separación. Las desigualdades que aparecen con más frecuencia son, con diferencia, las de cortes R-impares, seguidas de las desigualdades K-C. En el caso de las desigualdades Zigzag, el número de cortes hallados no es muy alto, pero hay que tener en cuenta que el algoritmo diseñado sólo es capaz de identificar desigualdades Zigzag pares que, como hemos visto en esta tesis, es un subconjunto muy pequeño de la familia completa de desigualdades Zigzag.

| | Óptimos (total/%) | Tiempo (seg.) | Nodos | Gap |
|---------|-------------------|---------------|-------|---------|
| CHR | 144 / 100 % | 0.2 | 0.2 | 0 % |
| HG | 53 / 100 % | 1.5 | 1.3 | 0 % |
| HD | 54 / 100 % | 1.4 | 1.3 | 0 % |
| ALB | 72 / 100 % | 2.3 | 1.5 | 0 % |
| MAD | 72 / 100 % | 9.8 | 2.7 | 0 % |
| A500 | 27 / 100 % | 39.5 | 6.5 | 0 % |
| A1000 | 27 / 100 % | 607.3 | 38.2 | 0 % |
| B500 | 24 / 100 % | 162.9 | 23.7 | 0 % |
| C750 | 21 / 87.5 % | 5183.9 | 143.9 | 0.027 % |
| D1000 | 20 / 87.3 % | 8972.9 | 282.0 | 0.410 % |
| GA500 | 27 / 100 % | 3279.9 | 57.1 | 0 % |
| GB500 | 19 / 79.2 % | 8463.0 | 193.4 | 0.607 % |
| M-Alba | 25 / 100 % | 1.7 | 4.6 | 0 % |
| M-Madr | 25 / 100 % | 508.5 | 101.1 | 0 % |
| M-Alda | 31 / 100 % | 58.8 | 25.2 | 0 % |
| MGB500 | 18 / 100 % | 2721.4 | 59.7 | 0 % |
| MGD1000 | 13 / 72.2 % | 10839.3 | 21.8 | 0.057 % |
| MRB500 | 18 / 100 % | 18.7 | 4.8 | 0 % |
| MRD1000 | 18 / 100 % | 1106.8 | 40.7 | 0 % |
| U-Alba | 15 / 100 % | 4.7 | 3.5 | 0 % |
| U-Madr | 15 / 100 % | 60.8 | 26.7 | 0 % |
| U-GRP | 10 / 100 % | 53.7 | 38.9 | 0 % |
| GTSP | 6 / 85.7 % | 5886.0 | 359.0 | 0.744 % |

Tabla 8.12: Prueba sobre todas las instancias

| | Con | R-imp | K-C | K-C ₀₂ | HC | HC ₀₂ | PB | ZZ | Tot |
|---------|-------|--------|-------|-------------------|------|------------------|------|-----|--------|
| CHR | 0.9 | 5.8 | 2.7 | 2.0 | 0.3 | 0.3 | 0.0 | 0.1 | 12.0 |
| HG | 2.2 | 30.6 | 8.2 | 5.3 | 1.6 | 0.6 | 0.3 | 0.6 | 49.3 |
| HD | 5.7 | 48.0 | 20.9 | 9.1 | 3.8 | 1.6 | 0.7 | 1.3 | 91.1 |
| ALB | 6.6 | 65.3 | 18.8 | 10.8 | 3.4 | 2.2 | 0.7 | 1.3 | 109.2 |
| MAD | 5.8 | 150.7 | 33.8 | 13.8 | 4.7 | 1.4 | 0.7 | 1.1 | 211.9 |
| A500 | 2.9 | 211.9 | 9.7 | 5.6 | 0.3 | 0.1 | 1.3 | 0.8 | 232.6 |
| A1000 | 1.5 | 600.9 | 10.6 | 14.0 | 0.3 | 0 | 2.0 | 0.4 | 629.8 |
| B500 | 22.3 | 1085.5 | 93.3 | 42.8 | 31.8 | 8.3 | 2.9 | 2.2 | 1289.0 |
| C750 | 33.6 | 2222.7 | 162.5 | 65.2 | 42.1 | 2.6 | 6.3 | 5.6 | 2540.7 |
| D1000 | 47.3 | 2935.6 | 226.5 | 79.9 | 27.1 | 3.3 | 23.6 | 5.0 | 3348.3 |
| GA500 | 25.1 | 294.3 | 27.5 | 7.8 | 0.7 | 0.0 | 5.6 | 0.1 | 361.3 |
| GB500 | 71.0 | 1479.8 | 187.2 | 66.2 | 41.0 | 3.3 | 23.4 | 2.6 | 1874.5 |
| M-Alba | 19.8 | 34.5 | 8.4 | 3.5 | 2.4 | 1.4 | 1.1 | 0 | 71.1 |
| M-Madr | 42.3 | 83.6 | 61.2 | 5.0 | 4.4 | 1.0 | 8.7 | 0.3 | 206.4 |
| M-Alda | 31.6 | 125.3 | 29.9 | 11.2 | 4.8 | 0.9 | 5.4 | 0.3 | 209.4 |
| MGB500 | 33.2 | 654.2 | 97.0 | 34.8 | 16.5 | 1.3 | 5.8 | 0.7 | 843.6 |
| MGD1000 | 46.1 | 875.1 | 101.1 | 48.9 | 12.2 | 1.2 | 9.4 | 1.8 | 1095.7 |
| MRB500 | 7.0 | 20.7 | 12.3 | 8.2 | 0.9 | 0.1 | 0.9 | 0.3 | 280.4 |
| MRD1000 | 14.4 | 1110.4 | 75.5 | 35.5 | 10.4 | 0.3 | 3.6 | 1.4 | 1251.5 |
| U-Alba | 21.2 | 75.9 | 17.3 | 11.2 | 3.8 | 3.6 | 2.1 | 0.1 | 135.1 |
| U-Madr | 27.2 | 222.8 | 68.0 | 16.5 | 12.9 | 4.4 | 3.7 | 2.3 | 357.7 |
| U-GRP | 35.6 | 55.6 | 172.0 | 56.0 | 85.8 | 41.7 | 1.9 | 1.0 | 449.6 |
| GTSP | 305.1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 305.1 |

Tabla 8.13: Cortes hallados

También puede observarse que, exceptuando el caso de las desigualdades de cortes R-impares, el número de desigualdades halladas es mucho más grande para la instancias generadas imitando casos reales que para las completamente aleatorias, lo que probablemente es debido a que las desigualdades K-C, Honeycomb y Path-Bridge y sus versiones 0-2 dependen mucho de la estructura del grafo.

Así pues, por todo lo comentado anteriormente, creemos que los resultados obtenidos son muy buenos. Nuestro algoritmo ha sido capaz de resolver instancias muy grandes del Windy General Routing Problem y, pese a no estar diseñado específicamente para ello, de algunos problemas particulares del mismo, como el GRP en grafos mixtos y no dirigidos y el TSP Gráfico. Aún queda, no obstante, trabajo por hacer. Algunos algoritmos de separación pueden ser mejorados, como en el caso de las desigualdades Zigzag, y otros nuevos pueden ser añadidos, como el de las desigualdades Path-Bridge₀₂.



Capítulo 9

Conclusiones

En esta tesis hemos estudiado el Problema General de Rutas con Viento (WGRP). Dicho problema consiste en, dado un grafo $G = (V, E)$, con un conjunto de vértices V , un conjunto de aristas E y dos costes c_{ij} , c_{ji} asociados a cada arista, uno por sentido de recorrido, y dados un subconjunto de vértices requeridos $V_R \subseteq V$ y un subconjunto de aristas requeridas $E_R \subseteq E$, hallar un tour de coste mínimo que recorra al menos una vez todas las aristas requeridas y visite todos los vértices requeridos. Este problema tiene un doble interés, ya que modeliza situaciones reales, de las cuales presentamos detalladamente un ejemplo en esta memoria, y, al mismo tiempo, generaliza a la mayoría de Problemas de Rutas por Arcos con un solo vehículo conocidos.

Hemos estudiado el poliedro asociado al espacio de soluciones del problema, describiendo las siguientes familias de desigualdades y demostrando que definen faceta del poliedro:

- Desigualdades triviales.
- Desigualdades de obligatoriedad.
- Desigualdades de conectividad.
- Desigualdades de cortes R-impares.
- Desigualdades K-C y K-C₀₂.
- Desigualdades Path-Bridge y Path-Bridge₀₂.
- Desigualdades Honeycomb y Honeycomb₀₂.

También hemos introducido un teorema de “lifting” que afirma que todas las desigualdades de configuración que inducen faceta para el WGRP en el grafo de configuración también inducen faceta del WGRP en el grafo original. Sin embargo, hemos visto que no todas las desigualdades que inducen faceta del poliedro del WGRP son desigualdades de configuración, por lo que hemos introducido el concepto de desigualdad de configuración débil. También hemos presentado una nueva familia de desigualdades válidas que inducen facetas para el WGRP, llamadas Zigzag, y que se engloban dentro de esta nueva categoría de desigualdades. Hemos estudiado la aplicación de estas nuevas desigualdades en otros problemas que son casos particulares del WGRP, así como bajo qué condiciones son válidas e inducen facetas en estos problemas.

Para la resolución del WGRP, hemos implementado un algoritmo de Ramificación y Corte (Branch & Cut), que incorpora, entre otros, un nuevo algoritmo heurístico de separación diseñado para la identificación de un cierto tipo de desigualdades Zigzag violadas. Hemos utilizado una serie de técnicas para mejorar la eficiencia del algoritmo Branch & Cut, como son la ramificación por restricciones, el *tailing-off*, el filtrado de desigualdades y el reinicio con cota superior artificial.

Este algoritmo de Ramificación y Corte ha sido probado sobre un gran número de instancias de muy diversas características. Algunas de ellas, las más grandes, han sido generadas aleatoriamente para este trabajo, mientras que otras han sido tomadas de la literatura existente. No sólo hemos utilizado instancias del WGRP, sino también de problemas que aparecen como casos particulares, como el WRPP, el MGRP, el MRPP, el GRP y el GTSP. Los tamaños de las instancias varían desde 7 a 1000 nodos, de 10 a las 4000 aristas y de 1 a 480 componentes R-conexas.

El algoritmo ha sido capaz de resolver en cuestión de segundos las instancias de tamaño pequeño y en algunos minutos las de tamaño mediano. La mayoría de las instancias de mayor tamaño también fueron resueltas óptimamente, y sólo para algunas de las instancias más difíciles y con mayor número de componentes R-conexas el algoritmo ha sido incapaz de llegar a la solución óptima en un tiempo límite de 10 horas, aunque el gap medio en estos conjuntos de instancias se mantiene por debajo del 0.7%.

Como un trabajo futuro nos hemos planteado la mejora del algoritmo de separación para las desigualdades Zigzag. Es nuestro objetivo diseñar un algoritmo que sea capaz de identificar desigualdades Zigzag impares violadas. También tenemos previsto diseñar un algoritmo Branch & Cut específico para el WPP que incorpore dicho algoritmo de separación. Al margen de las desigualdades R-impares, la única otra familia conocida hasta la fecha de desigualdades válidas que inducen faceta para el WPP es la de las desigualdades k -wheel de Win. Como hemos visto en esta tesis, las desigualdades 3-wheel son un caso particular de las desigualdades Zigzag impares. En cuanto al resto de desigualdades

k -wheel, creemos que las restrictivas condiciones necesarias para que sean válidas e induzcan faceta las hacen poco útiles en la práctica. Por tanto, pensamos que la incorporación de las desigualdades Zigzag impares puede tener un gran impacto en la resolución del WPP.

Por otra parte, planeamos estudiar a fondo la descripción del poliedro del WGRP asociado a grafos formados por cuatro nodos. Sería interesante comprobar si las desigualdades Zigzag son suficientes para obtener una descripción completa de dicho poliedro y, de no ser así, obtener nuevas familias de desigualdades válidas que permitan mejorar su descripción.

Bibliografía

- [1] D. Applegate, R.E. Bixby, V. Chvátal, and W. Cook. Finding cuts in the tsp. Technical report, DIMACS 95-05, Marzo 1995.
- [2] A.A. Assad and B.L. Golden. Arc routing methods and applications. In M. Ball et al., editor, *Handbooks in OR & MS, Vol. 8*. Elsevier Science, 1995.
- [3] P. Augerat, J.M. Belenguer, E. Benavent, A. Corberán, D. Naddef, and G. Rinaldi. Computational results with a branch-and-cut code for the capacitated vehicle routing problem. Technical Report RR949-M, ARTEMIS-IMAG, France, 1995.
- [4] A. Bachem and M. Grötschel. New aspects of polyhedral theory. *Modern Applied Mathematics: Optimization and Operations Research (B. Korte, eds)*. North Holland, Amsterdam, pages 51–106, 1982.
- [5] C. Balaguer, A. Giménez, J.M. Pastor, V.M. Padrón, and M. Abderrahim. A climbing autonomous robot for inspection applications in 3d complex environments. *Robotica*, 18:287–297, 2000.
- [6] M.O. Ball and M.J. Magazine. Sequencing of insertions in printed circuit board assembly. *Operations Research*, 36:192–201, 1988.
- [7] M. Bazaraa and J. Jarvis. *Linear Programming and Network Flows*. Wiley, New York, 1977.
- [8] E. Benavent, V. Campos, A. Corberán, and E. Mota. Problemas de rutas por arcos. *Qüestió*, 7:479–490, 1983.
- [9] E. Benavent, A. Carrota, A. Corberán, J.M. Sanchis, and D. Vigo. Lower bounds and heuristics for the windy rural postman problem. Technical report, Universidad de Valencia, 2003.
- [10] E. Benavent, A. Corberán, E. Piñana, I. Plana, and J.M. Sanchis. A scatter search algorithm for the windy rural postman problem. To appear in *Computers & Operations Research*, 2005.
- [11] C. Berge. *Graphs and Hypergraphs*. North Holland, Amsterdam, 1973.

-
- [12] L. Bodin and B. Golden. Classification in vehicle routing and scheduling. *Networks*, 11:97–108, 1981.
- [13] L. Bodin, B. Golden, A. Assad, and M. Ball. Routing and scheduling of vehicles and crews. *Computers and Operations Research*, 10:63–211, 1983.
- [14] U. Bondy, L. y Murty. *Graph Theory with Applications*. American Elsevier, New York, 1976.
- [15] V. Campos and J.V. Savall. A computational study of several heuristics for the DRPP. *Computational Optimization and Applications*, 4:67–77, 1995.
- [16] S. Chopra and G. Rinaldi. The graphical asymmetric traveling salesman polyhedron: Symmetric inequalities. *SIAM J. Discrete Math.*, 9(4):602–624, 1996.
- [17] N. Christofides. The optimum traversal of a graph. *Omega*, 1:719–732, 1973.
- [18] N. Christofides. *Graph Theory. An Algorithmic Approach*. Academic Press, New York, 1975.
- [19] N. Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. Technical report, Graduate School of Industrial Administration, Carnegie-Mellon University, 1976.
- [20] N. Christofides, V. Campos, A. Corberán, and E. Mota. An algorithm for the rural postman problem. Technical report, IC. OR. 81. 5., Imperial College, London, 1981.
- [21] N. Christofides, E. Benavent, V. Campos, A. Corberán, and E. Mota. An optimal method for the mixed postman problem. In P. Thoft-Christensen, editor, *System Modelling and Optimization*, Lectures Notes in Control and Information Sciences, page 59. Springer, Berlin, 1984.
- [22] N. Christofides, V. Campos, A. Corberán, and E. Mota. An algorithm for the rural postman problem on a directed graph. *Mathematical Programming Study*, 26:155–166, 1986.
- [23] V. Chvátal. *Linear Programming*. Freeman, New York, 1983.
- [24] A. Corberán and J.M. Sanchis. A polyhedral approach to the rural postman problem. *European Journal of Operational Research*, 79:95–114, 1994.
- [25] A. Corberán and J.M. Sanchis. The general routing problem polyhedron: Facets from the RPP and GTSP polyhedra. *European Journal of Operational Research*, 108:538–550, 1998.
- [26] A. Corberán, R. Martí, and A. Romero. Heuristics for the mixed rural postman problem. *Computers and Operations Research*, 27:183–203, 2000.

- [27] A. Corberán, A.N. Letchford, and J.M. Sanchis. A cutting plane algorithm for the general routing problem. *Mathematical Programming*, 90:291–316, 2001.
- [28] A. Corberán, R. Martí, and J.M. Sanchis. A grasp heuristic for the mixed chinese postman problem. *European Journal of Operational Research*, 142: 70–80, 2002.
- [29] A. Corberán, A. Romero, and J.M. Sanchis. The mixed general routing problem polyhedron. *Mathematical Programming*, 96(1):103–137, 2003.
- [30] A. Corberán, G. Mejía, and J.M. Sanchis. New results on the mixed general routing problem. To appear in *Computers & Operations Research*, 2005a.
- [31] A. Corberán, E. Mota, and J.M. Sanchis. A comparison of two different formulations for arc routing problems on mixed graphs. To appear in *Computers & Operations Research*, 2005b.
- [32] G. Cornuejols, J. Fonlupt, and D. Naddef. The traveling salesman problem on a graph and some related inequalities. *Mathematical Programming*, 33: 1–27, 1985.
- [33] G. Dantzig. *Linear Programming*. Freeman, New York, 1963.
- [34] M. Dror. *Arc Routing: Theory, Solutions and Applications*. Kluwer Academic Publishers, 2000.
- [35] J. Edmonds. The chinese postman problem. *Operations Research*, 13 Supplement 1:B73, 1965a.
- [36] J. Edmonds. Paths, trees and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965b.
- [37] J. Edmonds and E.L. Johnson. Matching, euler tours and the chinese postman. *Mathematical Programming*, 5:88–124, 1973.
- [38] H.A. Eiselt, M. Gendreau, and G. Laporte. Arc routing problems. part i: The chinese postman problem. *Operations Research*, 43(2):231–242, 1995a.
- [39] H.A. Eiselt, M. Gendreau, and G. Laporte. Arc routing problems. part ii: The rural postman problem. *Operations Research*, 43:399–414, 1995b.
- [40] L. Euler. The königsberg bridges. *Scientific American*, 189:66–70, 1953.
- [41] E. Fernández, O. Meza, R. Garfinkel, and M. Ortega. On the undirected rural postman problem: Tight bounds based on a new formulation. *Operations Research*, 51(2):281–291, 2003.
- [42] B. Fleischmann. A cutting plane procedure for the traveling salesman problem on road networks. *European Journal of Operational Research*, 21: 147–172, 1985.

- [43] B. Fleischmann. A new class of cutting planes for the symmetric traveling salesman problem. *Mathematical Programming*, 40:225–246, 1988.
- [44] H. Fleischner. Eulerian graphs and related topics. part i. In *Annals of Discrete Mathematics 45*. North-Holland, Amsterdam, 1990.
- [45] H. Fleischner. Eulerian graphs and related topics. part 2. In *Annals of Discrete Mathematics 50*. North-Holland, Amsterdam, 1991.
- [46] L.R. Ford and D.R. Fulkerson. *Flows in Networks*. Princeton University Press, NJ, 1962.
- [47] G.N. Frederickson. Approximation algorithms for some postman problems. *Journal of the Association for Computing Machinery*, 26:538–554, 1979.
- [48] R. Garfinkel and G. Nemhauser. *Integer Programming*. Wiley, New York, 1972.
- [49] G. Ghiani and G. Laporte. A branch-and-cut algorithm for the undirected rural postman problem. *Mathematical Programming*, 87(3):467–481, 2000.
- [50] M. Grötschel. Developments in combinatorial optimization. *Perspectives in Mathematics*, pages 249–294, 1984.
- [51] M. Grötschel and O. Holland. Solving matching problems with linear programming. *Mathematical Programming*, 3:243–259, 1985.
- [52] M. Grötschel and M.W. Padberg. Polyhedral theory. In E.L. Lawler et al, editor, *The Traveling Salesman Problem*, pages 251–305. J. Wiley & Sons, 1985.
- [53] M. Grötschel and Z. Win. A cutting plane algorithm for the windy postman problem. *Mathematical Programming*, 55:339–358, 1992.
- [54] M. Grötschel, M. Jünger, and G. Reinelt. A cutting plane for the linear ordering problem. *Operation Research*, 32:1195–1220, 1984.
- [55] M. Guan. Graphic programming using odd and even points. *Chinese Mathematics*, 1:273–277, 1962.
- [56] H. Gün. *Polyhedral Structures and Efficient Algorithms for Certain Classes of Directed Rural Postman Problems*. PhD thesis, Applied Math Program, University of Maryland, USA, 1993.
- [57] F. Harary. *Graph Theory*. Addison Wesley, Reading, MA, 1969.
- [58] A. Hertz, G. Laporte, and P. Nanchen. Improvement procedures for the undirected rural postman problem. *INFORMS Journal on Computing*, 11: 53–62, 1999.
- [59] K. Hoffman and M. Padberg. Lp-based combinatorial problem solving. *Annals of Op. Res.*, 4:145–194, 1985.

- [60] L. Khachian. A polynomial algorithm for linear programming. *Soviet Math. Doklady*, 20:191–194, 1979.
- [61] J.K. Lenstra and A. Rinnooy-Kan. On the general routing problem. *Networks*, 6:273–280, 1976.
- [62] J.K. Lenstra and A. Rinnooy-Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11(2):221–227, 1981.
- [63] A.N. Letchford. New inequalities for the general routing problem. *European Journal of Operational Research*, 96(2):317–322, 1997.
- [64] A.N. Letchford. The general routing problem: a unifying framework. *European Journal of Operational Research*, 112(1):122–133, 1999.
- [65] T.M. Liebling. Graphtheorie in planungs- und tourenproblemen. In *Lecture Notes in Operations Research and Mathematical Systems*, volume 21. Springer-Verlag, 1970.
- [66] E. Minieka. The chinese postman problem for mixed networks. *Management Science*, 25:643–648, 1979.
- [67] D. Naddef and G. Rinaldi. The symmetric traveling salesman polytope and its graphical relaxation: Composition of valid inequalities. *Mathematical Programming*, 51:359–400, 1991.
- [68] H. Nagamochi, T. Ono, and T. Ibaraki. Implementing an efficient minimum capacity cut algorithm. *Mathematical Programming*, 67:297–324, 1994.
- [69] G. Nemhauser and L. Wolsey. *Integer and Combinatorial Optimization*. Wiley, New York, 1988.
- [70] Y. Nobert and J.C. Picard. An optimal algorithm for the mixed chinese postman problem. *Networks*, 27:95–108, 1996.
- [71] C.S. Orloff. A fundamental problem in vehicle routing. *Networks*, 4:35–64, 1974.
- [72] M. Padberg and M. Rao. Odd minimum cut-sets and b-matchings. *Math. Oper. Res.*, 7:67–80, 1982.
- [73] M. Padberg and G. Rinaldi. A branch and cut algorithm for the resolution of larg-scale symmetric traveling salesman. *Mathematical Programming*, 47:19–36, 1990.
- [74] C. Papadimitriou. On the complexity of edge traversing. *J. ACM*, 23: 544–554, 1976.
- [75] C. Papadimitriou. The euclidean traveling salesman problem is np-complete. *Mathematical Programming*, 14(3):312–324, 1977.

-
- [76] C. Papadimitriou. Polytopes and complexity. *Progress in Combinatorial Optimization*, pages 295–305, 1984.
- [77] W. R. Pulleyblank. Polyhedral combinatorics. *Mathematical Programming: The State of the Art*, pages 312–345, 1983.
- [78] B. Raghavachari and J. Veerasamy. Approximation algorithms for mixed postman problem. In *Proceedings of the 6th Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 169–179, 1998.
- [79] T.K. Ralphs. On the mixed chinese postman problem. *Operations Research Letters*, 14:123–127, 1993.
- [80] G. Reinelt. TSPLIB - a traveling salesman problem library. *ORSA Journal on Computing*, 3:376–384, 1991.
- [81] G. Reinelt and D.O. Theis. About the general routing problem polytope. Technical report, Institute of Computer Science, Universidad de Heidelberg, 2004.
- [82] A. Romero. *El Problema del Cartero Rural en un Grafo Mixto*. PhD thesis, Universidad de Valencia, 1997.
- [83] J.M. Sanchis. *El Poliedro del Problema del Cartero Rural*. PhD thesis, Universidad de Valencia, 1990.
- [84] J.V. Savall. *Resultados Poliédricos y Algoritmos Aproximados para el DRPP*. PhD thesis, Universidad de Valencia, 1990.
- [85] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, Chchester, 1986.
- [86] Z. Win. *Contributions to Routing Problems*. PhD thesis, Universität Augsburg, Diciembre 1987.
- [87] Z. Win. On the windy postman problem on eulerian graphs. *Mathematical Programming*, 44:97–112, 1989.

Reunido el Tribunal que suscribe, en el día de la fecha,
acordó otorgar, por unanimidad, a esta Tesis doctoral de

D. Isaac Plana Andari
la calificación de Excel. leut. Cum laude

Valencia, a 8 de Julio de ~~19~~ 2005

El Secretario,

El Presidente

