



Universidad de Murcia

Facultad de Matemáticas

Trabajo Fin de Máster

Etiquetado óptimo de mapas

Máster en Matemática Avanzada

Alumno:

María de las Mercedes Pelegrín García
(mariamercedes.pelegrin@um.es)

Director académico:

Alfredo Marín Pérez
(amarin@um.es)

Murcia, Junio de 2016

Declaración de originalidad:

María de las Mercedes Pelegrín García, autora del TFM "Etiquetado óptimo de mapas", bajo la tutela del profesor Alfredo Marn Pérez, declara que el trabajo que presenta es original, en el sentido de que ha puesto el mayor empeño en citar debidamente todas las fuentes utilizadas.

A handwritten signature in blue ink, appearing to read "Alfredo", enclosed within a circular scribble.

En Murcia, a 4 de Julio de 2016.

Índice general

1. Introducción	1
1.1. La Optimización Combinatoria	1
1.2. El problema del etiquetado de mapas	5
1.3. Revisión de conceptos básicos	7
2. Etiquetado por empaquetamiento	11
2.1. Del etiquetado al empaquetamiento	11
2.2. Formulaciones del empaquetamiento de cardinal máximo	14
2.3. Algoritmo de optimización del empaquetamiento	15
2.3.1. Algoritmo de ramificación y corte	16
2.3.2. Estrategias para fijar variables	18
2.3.3. Desigualdades válidas y algoritmos de separación	23
3. Etiquetado por redes de restricción	41
3.1. Conceptos previos	42
3.2. Redes de etiquetado	43
3.3. Definiciones alternativas de los problemas de etiquetado	50
3.4. Formulaciones de optimización lineal entera	51
3.4.1. Formulación para ETI	52
3.4.2. Formulación para ETI extendida	53
3.4.3. Formulación para MNE	56
4. Una nueva aproximación	59
4.1. Modelos basados en clases de ambigüedad	60
4.2. Modelos con ordenación	62
4.3. Algunas propiedades	66

5. Conclusiones y trabajo futuro	71
5.1. Conclusiones	71
5.2. Trabajo futuro	72

Capítulo 1

Introducción

Este trabajo trata sobre el problema de optimización combinatoria correspondiente al etiquetado de mapas. En esta memoria se recogen varios modelos y procedimientos para resolver dicho problema, algunos existentes en la literatura y otros propuestos por Marín y Pelegrín. Los dos primeros capítulos presentan distintos modelados del problema de etiquetado, extraídos de las tesis doctorales de Verweij y Klau respectivamente. El primero de ellos sigue un enfoque más abstracto, trasladando el problema del etiquetado a un problema de Teoría de Grafos. El segundo es de carácter más geométrico y por tanto depende en mayor medida de la estructura física del problema. Por otro lado, el cuarto capítulo introduce una nueva aproximación al problema, que se basa en la ambigüedad del etiquetado para evaluar su calidad (Marín and Pelegrín, 2016). Finalmente, el último capítulo de este trabajo establece una serie de conclusiones y posibles extensiones futuras dentro de este ámbito de estudio.

En este primer capítulo se explica brevemente en qué consiste la Optimización Combinatoria; se presenta el problema del etiquetado de mapas; y, finalmente, se revisan algunos conceptos básicos relacionados con el algoritmo del símplex.

1.1. La Optimización Combinatoria

La Optimización Combinatoria es la rama de las Matemáticas que trata de encontrar, entre un conjunto finito de posibles soluciones cumpliendo una serie de restricciones, aquellas que maximizan o minimizan una determinada función. Todos los elementos mencionados (el conjunto de soluciones, las restricciones y la función) modelan un determinado problema con entidad propia. Dicho modelado se denomina *formulación* del problema. Para codificar una solución de la formulación, se emplean *variables de decisión*. Como el problema es de tipo combinatorio

estas variables (o un subconjunto de ellas) tomarán valores enteros. Las *restricciones* del problema se traducen en *desigualdades* que integran a las variables de decisión (o a un subconjunto de ellas). El fin último consiste en optimizar (minimizar o maximizar) una función, que viene dada de forma explícita como combinación de las variables de decisión, denominada *función objetivo*. El conjunto de valores de las variables de decisión que respetan todas las restricciones del problema forman la *región factible* del mismo. Una solución se dice factible si pertenece a la región factible del problema.

Para ilustrar los conceptos descritos a lo largo de esta sección, consideramos el problema de las n -reinas. Este es un problema bastante conocido que consiste en colocar n reinas en un tablero de ajedrez de tamaño $n \times n$ de manera que ninguna reina amenace a otra. Supongamos, para fijar ideas, que $n = 8$. Una solución al problema de las 8-reinas viene dada por una lista con 8 posiciones distintas del tablero. Definimos las siguientes variables binarias para cada $i, j \in \{1, \dots, 8\}$:

$$x_{i,j} = \begin{cases} 1 & \text{si en la posición } (i, j) \text{ del tablero hay una reina,} \\ 0 & \text{en otro caso.} \end{cases}$$

Para ser factible, una solución no puede incluir posiciones correspondientes a una misma fila, columna o diagonal del tablero, pues de lo contrario las correspondientes reinas se amenazarían entre sí. Como el número de filas y columnas del tablero coinciden con el número de reinas, sabemos que una solución factible del problema dispondrá exactamente una reina en cada fila y en cada columna del tablero. Para que en cada fila i , $i = 1, \dots, 8$, haya solo una reina deberá cumplirse que $\sum_{j=1}^8 x_{i,j} = 1$. De la misma forma, cada columna j puede ser ocupada por una única reina, es decir, $\sum_{i=1}^8 x_{i,j} = 1$ para cada $j = 1, \dots, 8$. Finalmente, debemos ocuparnos de las restricciones referentes a las diagonales del tablero. Observamos que dos posiciones cualesquiera (i_1, j_1) , (i_2, j_2) de una diagonal se caracterizan por que $|i_1 - i_2| = |j_1 - j_2|$. Por lo tanto debe cumplirse, para cada par de posiciones distintas $(i, j), (r, s) \in \{1, \dots, 8\} \times \{1, \dots, 8\}$ verificando $|i - r| = |j - s|$, que $x_{r,s} + x_{i,j} \leq 1$. De esta forma, a lo sumo una de las dos posiciones será ocupada por una reina. De momento hemos definido una representación de las soluciones del problema y una codificación de las restricciones. En este caso no existe una función que optimizar, cualquier distribución de las reinas que cumpla las condiciones dadas se considera igual de buena. Si, por ejemplo, queremos encontrar la solución donde la reina de la primera fila esté lo más a la izquierda posible, podemos minimizar la función $\sum_{j=1}^8 j \cdot x_{1,j}$. La formulación

completa del problema se escribiría de la siguiente forma:

$$\begin{aligned}
 \text{mín} \quad & \sum_{j=1}^8 j \cdot x_{1,j} \\
 \text{s.a} \quad & \sum_{j=1}^8 x_{i,j} = 1 \quad \forall i \in \{1, \dots, 8\}, \\
 & \sum_{i=1}^8 x_{i,j} = 1 \quad \forall j \in \{1, \dots, 8\}, \\
 & x_{r,s} + x_{i,j} \leq 1 \quad \forall i, j, r, s \in \{1, \dots, 8\} : |i - r| = |j - s|, (i, j) \neq (r, s) \quad (1.1) \\
 & x_{i,j} \in \{0, 1\} \quad \forall i, j \in \{1, \dots, 8\}.
 \end{aligned}$$

Además de modelar los problemas, la Optimización Combinatoria también se encarga de obtener la solución o soluciones a los modelos planteados. El hecho de que el conjunto de soluciones a un problema combinatorio sea finito puede llevar a conclusiones precipitadas respecto a la dificultad que entraña su resolución. En la inmensa mayoría de casos, el conjunto factible crece de manera exponencial con el tamaño del problema (en el ejemplo de las n -reinas hay $n^2 \cdot (n^2 - 1) \cdot \dots \cdot (n^2 - n + 1)$ elecciones posibles de n casillas del tablero). Esto hace inviable una resolución por exploración exhaustiva del conjunto factible, incluso si se incorporan estrategias que agilicen el proceso mediante el descarte de soluciones.

La Optimización Combinatoria engloba distintos métodos y estrategias, exactos o aproximados, para encontrar la solución o soluciones a este tipo de problemas de optimización. Estos procedimientos tienen una base matemática y son automatizables, de manera que se encuentran disponibles para su uso a través de una gran variedad de software específico. Algunos de estos procedimientos son los métodos de *ramificación y acotación*, los *hiperplanos de corte* o los *métodos de descomposición*. Muchos de estos métodos trabajan con la *relajación lineal* del problema en variables enteras. Esta relajación consiste en prescindir de las restricciones de integridad de las variables, es decir, permitir que estas puedan tomar cualquier valor real. Tal reducción conlleva el aumento del tamaño de la región factible, por lo que, si estamos maximizando (minimizando), una solución del problema relajado nunca será menor (mayor) que la solución óptima del problema original. Considerar el problema relajado nos permite por tanto obtener cotas sobre el valor objetivo.

Sin embargo, el modelado del problema y la resolución de la formulación resultante no son dos tareas completamente independientes, más bien todo lo contrario. Una buena formulación puede facilitar el trabajo del algoritmo de resolución, mientras que una mala formulación puede obstaculizar esta segunda tarea hasta el punto extremo de imposibilitarla completamente.

Consideremos de nuevo el ejemplo del problema de las n -reinas. El conjunto de restricciones (1.1) puede sustituirse por un conjunto menos numeroso de restricciones más fuertes. En lugar de comparar cada par de posiciones (i, j) y (r, s) que se encuentren en la misma diagonal, podemos agregar todas las restricciones de (1.1) en un grupo de restricciones más fuertes. Sea $D \subseteq \mathcal{P}(\{1, \dots, 8\} \times \{1, \dots, 8\})$ el conjunto de diagonales del tablero tanto en sentido ascendente como descendente. Entonces (1.1) pueden ser sustituidas por:

$$\sum_{(i,j) \in d} x_{i,j} \leq 1 \quad \forall d \in D. \quad (1.2)$$

Las restricciones (1.2) son más fuertes que las recogidas en (1.1) porque cualquier solución que las cumpla verificará también estas otras. Además, un tablero de tamaño $n \times n$ tiene $4n - 6$ diagonales ($2n - 3$ en cada sentido), por lo que (1.2) engloba $4n - 6$ restricciones. En cambio, el número de restricciones recogidas en (1.1) es del orden de n^2 .

Entonces, ¿qué se entiende en general por buena formulación? Las desigualdades de una formulación delimitan un poliedro en \mathbb{R}^n , siendo n el número de variables que intervienen en la misma. El *poliedro de la formulación* engloba a la región factible pero, al haber restricciones de integridad sobre las variables, no coincide con ella. Una formulación es mejor cuanto más se ajuste su poliedro a la región factible que delimita. En el mejor caso, no será posible encontrar una formulación cuyo poliedro contenga a la región factible del problema y esté a su vez estrictamente contenido en poliedro de la relajación lineal de la formulación actual. Este caso se da cuando las desigualdades de la formulación conforman la *envolvente convexa de la región factible*. En la mayoría de los problemas esta envolvente es demasiado compleja para ser representada de forma explícita. En su lugar, se busca añadir a la formulación desigualdades que no puedan ser mejoradas, y que delimitarán parte de esta envolvente convexa. Como la mayoría de los métodos de resolución trabajan con relajaciones del problema, es importante que la región factible correspondiente esté delimitada de la forma más precisa posible, para descartar así tantos puntos como sea posible. Sin embargo, una mayor precisión a la hora de delimitar esta región suele traducirse en un aumento del número de desigualdades de la formulación. Este incremento puede hacer más lento el procedimiento de resolución, el cual tendrá que almacenar y manejar más información. Como consecuencia, se busca un equilibrio entre precisión y sobrecarga del procedimiento. En este contexto surgen los llamados *algoritmos de separación*, que establecen estrategias para añadir o eliminar desigualdades durante el proceso de resolución, a medida que estas sean o no necesarias.

1.2. El problema del etiquetado de mapas

Un problema primordial en cartografía es el de distribuir etiquetas que describan algunas zonas destacadas en un mapa, con calidad suficiente para que este pueda ser leído. Estas zonas distinguidas son principalmente de tres tipos: puntos, representando ciudades, estaciones, hoteles, nodos de un grafo, etcétera; líneas, que pueden ser calles, carreteras, ríos, aristas de un grafo, etcétera; y áreas, representando países, distritos, caras de un grafo planar... Una vez que se determina el conjunto de zonas que se quieren representar, existen dos formas de componer el mapa. La primera posibilidad es localizar las zonas distinguidas de forma más o menos arbitraria para conseguir un mapa armonioso, sin tener en cuenta la posición geográfica que estas zonas ocupan en la realidad. Ejemplos clásicos de esta práctica son los mapas de metro, donde las estaciones se disponen a lo largo de líneas rectas unidas formando ángulos de 45 o 90 grados, encontrándose a la misma distancia aquellas que son consecutivas. La segunda posibilidad es distribuir las zonas destacadas de acuerdo con su posición geográfica. En este caso, tras identificar las zonas, estas necesitan ser etiquetadas, esto es, se escribirá el nombre o la descripción de cada zona en algún lugar alrededor de la misma. A simple vista debería poder apreciarse qué etiqueta se corresponde con qué zona. Etiquetas colocadas siguiendo alguna estrategia intuitiva se solaparán en la mayoría de casos. Incluso si no lo hacen, pueden producir ambigüedad debido a su posición respecto al resto de zonas distinguidas.

La creciente cantidad de datos para la cual se producen gráficos informativos lleva a aumento de la necesidad de desarrollar procedimientos automáticos de etiquetado. Dada la complejidad de los problemas subyacentes a esta tarea, el etiquetado manual es una opción tediosa y costosa en tiempo. Además del clásico mapa cartográfico, el problema del etiquetado surge en sistemas de información geográfica, de navegación, interfaces gráficas u otro tipo de sistemas automáticos donde el diseño manual es imposible. Los numerosos campos de aplicación atraen a muchos investigadores en ciencias de la computación y matemáticas. Pero no se trata sólo de un tema de estudio atractivo desde un punto de vista práctico, sino que en esta área aparecen diversos problemas de optimización combinatoria con interesantes propiedades matemáticas. Existen varios criterios que han sido considerados para decidir cuándo un etiquetado es de buena calidad:

- En un buen mapa la distribución de las etiquetas debe ser tan poco ambigua como sea posible. Es decir, debe parecer claro para el lector qué etiqueta corresponde a qué zona destacada del mapa.

- La información dispuesta en las etiquetas debe ser legible. La falta de ambigüedad por sí sola no ayuda si el usuario no puede leer las etiquetas.
- Ninguna o solo unas pocas etiquetas se solapan. Evidentemente, los solapamientos disminuyen la legibilidad del mapa.

Existen diferentes tipos de modelos de etiquetado, según las opciones consideradas a la hora de situar las etiquetas en el mapa. Si para cada punto existe un número finito de formas de colocar su etiqueta, surge un problema de tipo combinatorio, mientras que si hay infinitas posibilidades obtendremos un modelo continuo. En el primer caso se consideran varias posiciones fijadas de antemano para las etiquetas. En el segundo caso, las etiquetas pueden ser deslizadas de manera continua en determinadas direcciones alrededor del punto.

En este trabajo, supondremos que las zonas destacadas que se desean etiquetar son puntos en un mapa bidimensional. Los otros dos casos en que estas zonas son líneas o áreas se reducen al caso considerado determinando de antemano un punto perteneciente a la zona, alrededor del cual se colocará la etiqueta. A lo largo de este trabajo consideraremos $\mathcal{P} = \{p_1, \dots, p_n\}$ el conjunto de puntos a etiquetar y llamaremos $I = \{1, \dots, n\}$ al conjunto de índices correspondientes. Para cada $i \in I$, L_i será el conjunto de etiquetas posibles del punto, y nos referiremos con $L = \cup_{i \in I} L_i$ al conjunto de todas las etiquetas del modelo. Un elemento $\ell \in L$ no representa una etiqueta como objeto abstracto sino que se refiere a la región física del plano que ocupará la etiqueta asociada a un determinado punto. Por ejemplo, supongamos que el punto p_i representa la ciudad de Murcia y tenemos dos posiciones donde ubicar la etiqueta. En este caso el conjunto L_i estaría formado por dos etiquetas, que serían los rectángulos del plano correspondientes a las dos ubicaciones posibles de la cadena “Murcia”. Una instancia concreta de un problema de etiquetado $P = (\mathcal{P}, \{L_i\}_{i \in I})$ viene determinada por el conjunto de todos los puntos con sus posibles etiquetas.

A continuación se da una definición formal del concepto de etiquetado y del número de solapamientos.

Definición 1.2.1. *Dada una instancia $(\mathcal{P}, \{L_i\}_{i \in I})$ del problema del etiquetado, un etiquetado ρ para $(\mathcal{P}, \{L_i\}_{i \in I})$ es una aplicación $\rho : I \rightarrow \cup_{i \in I} L_i$, tal que $\rho(i) \in L_i \forall i \in I$.*

Definición 1.2.2. *Dada una instancia $(\mathcal{P}, \{L_i\}_{i \in I})$ del problema del etiquetado y un etiquetado*

ρ el conjunto de pares de etiquetas solapadas de ρ es

$$O(\rho) = \{(i, k) \in I \times I : \rho(i) \cap \rho(k) \neq \emptyset, i \neq k\}.$$

De acuerdo con los criterios de calidad de un mapa anteriormente mencionados, se definen los siguientes problemas de etiquetado.

Definición 1.2.3. (Problema del etiquetado puro, ETI). Dada una instancia $(\mathcal{P}, \{L_i\}_{i \in I})$ del problema del etiquetado, encontrar, si existe, un etiquetado ρ para $(\mathcal{P}, \{L_i\}_{i \in I})$ de manera que $O(\rho) = \emptyset$.

Definición 1.2.4. (Problema de maximización del número de etiquetas, MNE). Dada una instancia $(\mathcal{P}, \{L_i\}_{i \in I})$ del problema del etiquetado, encontrar el mayor subconjunto $I' \subseteq I$ y un etiquetado ρ para el problema $(\mathcal{P}', \{L_i\}_{i \in I'})$ con $O(\rho) = \emptyset$, $\mathcal{P}' = \{p_i : i \in I'\}$.

Definición 1.2.5. (Problema de minimización del número de solapamientos, MNS). Dada una instancia $(\mathcal{P}, \{L_i\}_{i \in I})$ del problema del etiquetado, encontrar un etiquetado ρ para $(\mathcal{P}, \{L_i\}_{i \in I})$ de manera que $|O(\rho)|$ sea mínimo.

Definición 1.2.6. (Problema de maximización del tamaño de etiqueta, MTE). Dada una instancia $(\mathcal{P}, \{L_i\}_{i \in I})$ del problema del etiquetado, encontrar el mayor factor $\alpha \in \mathbb{Q}$ y un etiquetado ρ para el problema $(\mathcal{P}, \{L'_i\}_{i \in I})$ con $O(\rho) = \emptyset$. Aquí, las etiquetas de L'_i se corresponden con las de L_i con un factor de escala $1 : \alpha$.

1.3. Revisión de conceptos básicos

En esta sección vamos a revisar algunos conceptos relacionados con el *algoritmo del símplex*. Este algoritmo se emplea para resolver problemas de optimización donde las variables son positivas y las restricciones lineales. Algunos de los conceptos que se presentan aquí serán utilizados en los siguientes capítulos, ya que se trabajará con relajaciones lineales del problema de etiquetado.

A lo largo de la sección trabajaremos con $\mathcal{P} \equiv \min\{cx : Ax = b, x \geq 0\}$ un problema de optimización en forma estándar. Supondremos que n es el número de variables y m el número de restricciones. La matriz A es una matriz $m \times n$ que reúne los coeficientes de m restricciones de igualdad referentes a las n variables del problema. El vector $b \in \mathbb{R}^m$ se denomina vector de recursos, mientras que $c \in \mathbb{R}^n$ es el vector de costes. Obsérvese que cualquier problema lineal

puede escribirse según la forma estándar, multiplicando las restricciones “ \geq ” por menos uno, añadiendo variables de holgura en el lado izquierdo de las restricciones “ \leq ” y/o multiplicando las variables negativas por menos uno.

Definición 1.3.1. *Una matriz básica factible B es una submatriz de A cuyo determinante es distinto de cero: $B_{m \times m} \subset A$, $|B| \neq 0$. Si además se cumple $B^{-1}b \geq 0$ entonces B se dice básica factible.*

Sea $B_{m \times m} \subset A$ una submatriz básica factible. Si descomponemos A en dos submatrices, $A = (B, N)$, también podemos descomponer la igualdad $Ax = b$ en $Bx_B + Nx_N = b$. Como B es básica existe B^{-1} , por lo que la desigualdad anterior se escribe como $x_B = B^{-1}b - B^{-1}Nx_N$. Tomando $x_B = B^{-1}b$ y $x_N = 0$ obtenemos una solución factible de \mathcal{P} (cumple la igualdad $Ax = b$ y además $x \geq 0$ por ser B factible). Por lo tanto, cada matriz básica factible produce una solución factible del problema, denominada solución básica factible.

Es un resultado bien conocido en el campo de la Optimización Lineal que las soluciones óptimas de cualquier problema \mathcal{P} son soluciones básicas factibles. El algoritmo del s implex consiste en, partiendo de una soluci on b asica factible, modificarla para obtener otra soluci on factible asociada a una base distinta, hasta encontrar aquella que sea  optima. Para saber cu ando una soluci on es  optima, en cada paso se comprueba si se cumple el criterio de optimalidad. Si B es la matriz b asica factible en un paso determinado del algoritmo, este criterio es:

$$c_B B^{-1} a_{.j} - c_j \leq 0 \quad \forall j \in N \text{ columna (variable) no b asica.} \quad (1.3)$$

En general, para $x \in \mathbb{R}^n$ cualquiera se tiene:

$$\begin{aligned} cx &= c_B x_B + c_N x_N = c_B (B^{-1}b - B^{-1}N x_N) + c_N x_N = c_B B^{-1}b - (c_B B^{-1}N - c_N) x_N = \\ &= c_B B^{-1}b - \sum_{j \in N} (c_B B^{-1} a_{.j} - c_j) x_j. \end{aligned} \quad (1.4)$$

Por lo tanto, si se cumple el criterio (1.3), para cualquier $x \in \mathbb{R}^n$ se tendr a $cx \geq c_B B^{-1}b$. Por otro lado, para la soluci on b asica \bar{x} asociada a B se cumple:

$$c\bar{x} = c_B B^{-1}b - \sum_{j \in N} (c_B B^{-1} a_{.j} - c_j) \bar{x}_j = c_B B^{-1}b. \quad (1.5)$$

En consecuencia, \bar{x} es  optima. Las cantidades $c_B B^{-1} a_{.j} - c_j$ se denominan costes reducidos y representan el cambio en la funci on objetivo debido a la variaci on en una unidad de la variable

x_j . Para comprender esta interpretación basta observar la expresión de la función objetivo para una solución cualquiera (1.4) junto con el valor de la función objetivo en la solución actual asociada a la base factible (1.5).

Capítulo 2

Etiquetado por empaquetamiento

En su tesis doctoral así como en el informe técnico asociado (Verweij, 2000; Strijk et al., 2000), Verweij identifica el problema de maximizar el número de puntos de un mapa etiquetados sin solapamiento (es decir, el problema MNE) con el de encontrar el empaquetamiento de nodos de máximo cardinal en un grafo. En su estudio, Verweij determina el grafo asociado al problema del etiquetado y desarrolla un algoritmo de ramificación y corte (o *branch and cut*) para resolver el problema de forma exacta. La originalidad del algoritmo reside en las estrategias seguidas para fijar el mayor número posible de variables en cada paso, así como en el procedimiento de separación que gestiona la adición de nuevas desigualdades. Este capítulo se centra en plasmar estos dos aspectos, que, desde un punto de vista teórico, constituyen la principal aportación de la tesis de Verweij al problema del etiquetado.

2.1. Del etiquetado al empaquetamiento

En esta sección veremos cómo trasladar el problema del etiquetado al contexto de la Teoría de Grafos, para finalmente identificarlo con el problema del empaquetamiento de nodos. Pero antes de presentar el resultado que da nombre a esta sección, es necesario introducir algunas nociones básicas de Teoría de Grafos.

Definición 2.1.1. *Un grafo no dirigido $G = (V, E)$ es una estructura formada por un conjunto finito de nodos, V , y un conjunto de aristas o pares no ordenados de nodos adyacentes, $E \subseteq \{e = (v, w) | v, w \in V, v \neq w\}$.*

Definición 2.1.2. *Un grafo bipartito $G = (V, E)$ es aquel para el que existe una partición de nodos $V = V^1 \dot{\cup} V^2$ de forma que si $(u, v) \in E$ entonces $u \in V^1$ y $v \in V^2$ o viceversa.*

Definición 2.1.3. Dado un grafo $G = (V, E)$ y un nodo $v \in V$ se define el vecindario de v como el conjunto de nodos adyacentes a v :

$$N(v) = \{w \in V | (v, w) \in E\}.$$

Definición 2.1.4. Sea un grafo $G = (V, E)$ y un subconjunto de nodos $S \subseteq V$. El grafo con conjunto de nodos S y conjunto de aristas $E(S) = \{(u, v) \in E | u, v \in S\}$ es el grafo inducido por S y se denota como $G[S] = (S, E(S))$.

Definición 2.1.5. Un paseo de v_0 a v_k en G es una sucesión finita de nodos, (v_0, v_1, \dots, v_k) , de manera que cada uno es adyacente al anterior. Cuando $v_0 = v_k$, el paseo se dice cerrado. Diremos que un paseo tiene longitud impar (par) si k es impar (par).

Definición 2.1.6. Un camino (v_0, v_1, \dots, v_k) es un paseo en el que los nodos son distintos entre sí. Cuando $v_0 = v_k$, siendo el resto de nodos distintos entre sí y distintos de v_0 , a la sucesión (v_0, v_1, \dots, v_k) se le llama ciclo.

Definición 2.1.7. Una cuerda de un ciclo (paseo) es una arista que incide en dos nodos no consecutivos del ciclo (paseo).

Definición 2.1.8. Un grafo $G = (V, E)$ se dice conexo si existe un camino entre cualquier par de nodos.

Definición 2.1.9. Una componente conexa de un grafo $G = (V, E)$ es un grafo conexo inducido $G[S]$ maximal, es decir no existe un subconjunto de nodos $S' \subseteq V$ tal que $G[S']$ es conexo y $S \subseteq S'$.

Definición 2.1.10. Un empaquetamiento de nodos de un grafo $G = (V, E)$ es un subconjunto de nodos $V' \subseteq V$ no adyacentes:

$$\forall v, w \in V', (v, w) \notin E.$$

Definición 2.1.11. Dado un grafo $G = (V, E)$, un subconjunto de nodos $C \subseteq V$ se llama clique si cualquier par de nodos son adyacentes:

$$\forall c_1, c_2 \in C, (c_1, c_2) \in E.$$

Consideremos ahora una instancia del problema de etiquetado. Sea $\mathcal{P} = \{p_1, \dots, p_n\}$ un conjunto de puntos en el plano y $L_i, i \in I = \{1, \dots, n\}$, el conjunto de posibles ubicaciones de

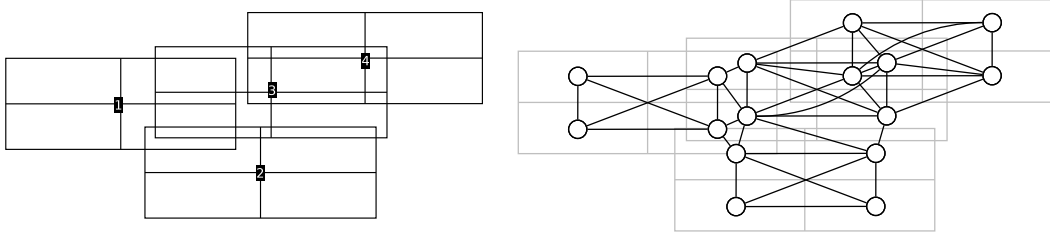


Figura 2.1: Una instancia $(\mathcal{P}, \{L_i\}_{i \in I})$ y su grafo intersección $G_L = (V_L, E_L)$

la etiqueta de cada punto. El par $(\mathcal{P}, \{L_i\}_{i \in I})$ determina un problema de etiquetado de mapas. Llamaremos $L = \cup_{i \in I} L_i$ al conjunto de etiquetas en el plano asociadas a la instancia anterior.

Definición 2.1.12. *En las condiciones anteriores, el grafo intersección asociado a una instancia del problema de etiquetado, $G_L = (V_L, E_L)$, se define como:*

$$V_L = \{v_\ell | \ell \in L\}, \quad (2.1)$$

$$E_L = \{(v_j, v_\ell) | j \neq \ell, j \cap \ell \neq \emptyset\}. \quad (2.2)$$

En la definición anterior, los elementos de L son regiones físicas en el plano, por lo que tienen sentido las intersecciones de la condición (2.2). La condición (2.1) garantiza que existe un nodo por cada etiqueta de L , mientras que (2.2) asegura que existirá una arista entre dos nodos cuyas correspondientes etiquetas tengan intersección no vacía. La Figura 2.1 muestra un ejemplo gráfico de la correspondencia entre una instancia del problema de etiquetado y su grafo intersección.

Si suponemos que cada punto pertenece a la región que ocupa su etiqueta, es decir, $p_i \in \ell \forall \ell \in L_i$, por la condición (2.2), para cada $i \in I$ los nodos correspondientes a L_i formarán un clique en G_L . A lo largo del capítulo se asumirá que esto ocurre así.

Teorema 2.1.1. *Sea $(\mathcal{P}, \{L_i\}_{i \in I})$ una instancia del problema de etiquetado de mapas. Existe una biyección entre el conjunto de las soluciones factibles de MNE para la instancia $(\mathcal{P}, \{L_i\}_{i \in I})$ y los empaquetamientos de nodos del grafo intersección G_L .*

Demostración. Recordamos que la formulación MNE trata de encontrar el mayor subconjunto $I' \subseteq I$ y un etiquetado ρ para la instancia $(\mathcal{P}', \{L_i\}_{i \in I'})$ con $O(\rho) = \emptyset$, $\mathcal{P}' = \{p_i : i \in I'\}$. Supongamos entonces que $I' \subseteq I$ junto con el etiquetado $\rho : I' \rightarrow L$ es una solución de MNE. Sea $L' \subseteq L$ el conjunto de etiquetas de la solución, i.e.,

$$L' = \{\ell \in L | \ell = \rho(i), \text{ para algún } i \in I'\},$$

y consideremos $V' \subseteq V_L$ el conjunto de nodos asociados, i.e.

$$V' = \{v_\ell \in V_L \mid \ell \in L'\}.$$

Como I' es una solución de MNE, para cada par de etiquetas $\ell, j \in L'$, $\ell \neq j$, se cumple que $\ell \cap j = \emptyset$. Por tanto, $(v, w) \notin E_L \forall v, w \in V'$, lo cual implica que V' es un empaquetamiento de nodos de G_L .

Supongamos ahora que $V' \subseteq V_L$ es un empaquetamiento de nodos. Podemos definir $I' = \{i \in I \mid v_\ell \in V' \text{ para algún } \ell \in L_i\}$ y $\rho : I' \rightarrow L$ tal que $\rho(i) = \ell$ con $v_\ell \in V'$. La definición anterior es correcta puesto que, al ser V' un empaquetamiento, para cada $i \in I'$ existe un único $\ell \in L_i$ tal que $v_\ell \in V'$ (los nodos asociados a etiquetas de L_i forman un clique). Además, por ser V' un empaquetamiento $(v_\ell, v_j) \notin E_L \forall v_\ell, v_j \in V'$, es decir, $O(\rho) = \emptyset$. Como consecuencia I' junto con el etiquetado ρ es una solución factible de MNE. \square

Como corolario del teorema anterior, es equivalente encontrar una solución al problema de etiquetado $(\mathcal{P}, \{L_i\}_{i \in I})$ con máximo número de etiquetas que encontrar un empaquetamiento de nodos de G_L de cardinal máximo. Es decir,

$$\max\{|\mathcal{S}| : \mathcal{S} \text{ es una solución de } (\mathcal{P}, \{L_i\}_{i \in I})\} \equiv \max\{|\mathcal{S}| : \mathcal{S} \subseteq V_L \text{ es un empaquetamiento}\}.$$

Una vez transformado el problema inicial en un problema de empaquetamiento, nos centraremos en el análisis de este último en un sentido general. En adelante, nos abstraeremos del contexto del etiquetado para trabajar sobre un grafo cualquiera $G = (V, E)$ (que en nuestro caso se identificará con el grafo intersección).

2.2. Formulaciones del empaquetamiento de cardinal máximo

Para obtener una formulación de optimización lineal entera del problema de encontrar el empaquetamiento de mayor cardinal en un grafo $G = (V, E)$, consideramos las variables $x_v \in \{0, 1\} \forall v \in V$. Un nodo $v \in V$ pertenecerá al empaquetamiento si y sólo si $x_v = 1$. Sea el conjunto:

$$P_E = \{\mathbf{x} \in \mathbb{R}^{|V|} : x_u + x_v \leq 1 \forall (u, v) \in E, \mathbf{x} \geq \mathbf{0}\};$$

entonces el problema del empaquetamiento de cardinal máximo se puede formular como:

$$\text{máx} \left\{ \sum_{v \in V} x_v \mid \mathbf{x} \in P_E \cap \{0, 1\}^{|V|} \right\}.$$

Frente a esta formulación, a la que nos referiremos como *formulación en aristas* se puede establecer una *formulación en cliques*. Sea $C \in V$ un clique de G , la desigualdad

$$\sum_{v \in C} x_v \leq 1$$

es verificada por cualquier $\mathbf{x} \in P_E$. Dada una colección de cliques \mathcal{C} en G (no necesariamente maximales), de manera que para cada arista $(u, v) \in E$ existe un clique $C \in \mathcal{C}$ con $\{u, v\} \subseteq C$, definimos

$$P_{\mathcal{C}} = \left\{ \mathbf{x} \in \mathbb{R}^{|V|} : \sum_{v \in C} x_v \leq 1 \forall C \in \mathcal{C}, \mathbf{x} \geq \mathbf{0} \right\}.$$

Como los cliques de \mathcal{C} “cubren” las aristas de E , tenemos que $P_{\mathcal{C}} \subseteq P_E$. Por otro lado, la inclusión $P_E \subseteq P_{\mathcal{C}}$ solo se cumple si las variables son binarias. Si las sumas de variables correspondientes a los nodos de un clique dos a dos son menores o iguales que uno, la suma de todas las variables correspondientes a este clique no tiene por qué ser menor o igual que uno en el caso general (por ejemplo, tómesese un clique de 3 nodos con $x_1 = x_2 = x_3 = 0.5$). La formulación en cliques del problema del empaquetamiento es:

$$\text{máx} \left\{ \sum_{v \in V} x_v \mid \mathbf{x} \in P_{\mathcal{C}} \cap \{0, 1\}^{|V|} \right\}.$$

Cuando \mathcal{C} es una colección de cliques maximales, nos referiremos a esta formulación como *formulación en cliques maximales*.

2.3. Algoritmo de optimización del empaquetamiento

En Strijk et al., 2000, se utiliza un algoritmo de ramificación y corte para buscar una solución óptima al problema del empaquetamiento de cardinal máximo. Este algoritmo utiliza características específicas del problema para mejorar la eficiencia del proceso de búsqueda de la solución óptima. Sin embargo, antes de describir las estrategias que se siguen para esta mejora, es necesario introducir el concepto de algoritmo de ramificación y corte.

2.3.1. Algoritmo de ramificación y corte

Un algoritmo de ramificación y corte es la combinación de un procedimiento de ramificación y acotación (*branch and bound*) con estrategias basadas en hiperplanos de corte. A continuación, se proporciona una descripción básica de ambos algoritmos, de los que puede consultarse información más detallada en Wolsey, 1998.

Ramificación y acotación

Consideremos el problema $\Pi : z_{opt} = \max\{z(\mathbf{x}) : \mathbf{x} \in P, \mathbf{x} \in \mathbb{Z}^n\}$, donde z es una función lineal y $P \subseteq \mathbb{R}^n$ un poliedro. Consideremos por otro lado la relajación lineal $\bar{\Pi} : \bar{z} = \max\{z(\mathbf{x}) : \mathbf{x} \in P\}$. Como la región factible de la relajación $\bar{\Pi}$ es mayor que la del problema original, se cumple que $\bar{z} \geq z_{opt}$.

El algoritmo de ramificación y acotación trabaja con $\bar{\Pi}$ y con subproblemas suyos, dispuestos en un árbol. Diremos que $\bar{\Pi}_1 : \max\{z(\mathbf{x}) : \mathbf{x} \in P_1\}$ es subproblema de $\bar{\Pi}_2 : \max\{z(\mathbf{x}) : \mathbf{x} \in P_2\}$ si $P_1 \subseteq P_2$. Cada nodo del árbol se corresponderá con un problema, que será un subproblema del problema situado en el nodo padre, es decir, el nodo que le precede en el árbol. En el nodo raíz del árbol se encuentra la relajación lineal del problema original, $\bar{\Pi}$.

El algoritmo va construyendo el árbol de ramificación de manera que en una etapa k tendremos una colección de problemas, $\bar{\Pi}_1, \dots, \bar{\Pi}_\ell$, cuyos poliedros asociados P_1, \dots, P_ℓ son disjuntos dos a dos y la unión de todos ellos contiene a cualquier vector entero del poliedro inicial P . La única salvedad para esta última afirmación es que el vector haya sido descartado por encontrarse en una rama del árbol que no supone mejora. Esto es, un nodo del árbol puede descartarse si el valor óptimo del subproblema asociado está acotado superiormente por un valor que es menor que el correspondiente a la mejor solución entera encontrada hasta el momento. En tal caso, dicho nodo no será ramificado, es decir, no se explorarán sus subproblemas. A esta estrategia se le conoce como poda del árbol.

El algoritmo de ramificación y acotación consiste en lo siguiente. En cada paso, se mantiene un conjunto de problemas abiertos (sin resolver), la mejor solución (entera) de Π encontrada hasta el momento, x^* , y su valor objetivo, $z^* = z(x^*)$. Al comienzo del algoritmo, el único problema abierto es $\bar{\Pi}$. En cada iteración i , se selecciona un problema $\bar{\Pi}_i$ del conjunto de problemas abiertos y se resuelve. Sea x^i la solución óptima obtenida y $z^i = z(x^i)$ el valor óptimo asociado. Si el problema es infactible o $z^i \leq z^*$, $\bar{\Pi}_i$ se elimina del conjunto de problemas abiertos y se realiza una nueva iteración. Nótese que z^i es una cota superior para el valor óptimo de

cualquier subproblema de $\bar{\Pi}_i$, en particular, cualquier solución entera de un subproblema suyo tendrá un valor objetivo menor o igual que z^i . Por lo tanto, en este primer caso no es necesario expandir el nodo correspondiente al problema $\bar{\Pi}_i$. En el caso en que x^i sea entero, si $z^i > z^*$, actualizamos los valores x^* y z^* , $x^* = x^i$ y $z^* = z^i$, eliminamos $\bar{\Pi}_i$ del conjunto de problemas abiertos y continuamos con la siguiente iteración. Si no se da ninguno de los casos anteriores tenemos x^i una solución fraccional de $\bar{\Pi}_i$. Entonces, identificamos una componente j con x_j^i no entero y generamos dos subproblemas $\bar{\Pi}_i^1$ y $\bar{\Pi}_i^2$ añadiendo respectivamente las restricciones $x_j \leq \lfloor x_j^i \rfloor$ y $x_j \geq \lceil x_j^i \rceil$ a P_i . Si x_j es una variable binaria se añaden respectivamente las restricciones $x_j = 0$ y $x_j = 1$. Conviene resaltar que x^i no pertenece a P_i^1 ni a P_i^2 . De esta manera, reemplazamos $\bar{\Pi}_i$ por $\bar{\Pi}_i^1$ y $\bar{\Pi}_i^2$ en el conjunto de problemas abiertos y continuamos con la siguiente iteración. El proceso terminará cuando el conjunto de subproblemas abiertos quede vacío, siendo en ese momento x^* y z^* la solución y valor óptimo del problema Π .

Hiperplanos de corte

Para que el procedimiento anterior funcione de manera eficiente, es conveniente obtener buenas cotas superiores e inferiores del valor óptimo de Π , ya que estas cotas son útiles para la poda del árbol. Formulaciones más fuertes (i.e. más restrictivas) del problema nos darán mejores cotas. En general, la formulación Π puede ser reforzada añadiendo desigualdades válidas al conjunto de restricciones que corten al poliedro P , es decir, desigualdades que cumplan todos los puntos enteros del poliedro pero violadas por algunos puntos fraccionales de P . Los algoritmos de planos de corte tratan de reforzar las formulaciones siguiendo esta idea. Se comienza resolviendo la relajación $\bar{\Pi}$. Si la solución es entera se termina el procedimiento. En otro caso, se inspeccionan una serie de desigualdades válidas del problema que han sido previamente identificadas, comprobando cuáles de ellas son violadas por la solución. Las que sean violadas (o un subconjunto de ellas) son añadidas al poliedro P , y se repite el procedimiento para el problema generado.

Un algoritmo ramificación y corte construye un árbol de subproblemas, de manera similar al algoritmo de ramificación y acotación. En cada nodo del árbol, en primer lugar se ejecuta un procedimiento basado en hiperplanos de corte. Una vez se ejecuta el número deseado de iteraciones del algoritmo de hiperplanos de corte, se toma el problema reforzado resultante para aplicar los pasos correspondientes al algoritmo de ramificación.

2.3.2. Estrategias para fijar variables

En nuestro caso, el algoritmo de ramificación y acotación trabajará con variables binarias. Por lo tanto, la ramificación de los nodos del árbol consistirá en fijar variables a 0 o a 1. Con el fin de acelerar el procedimiento lo máximo posible, en lugar de ramificar en una sola componente del vector de variables, los autores Strijk et al., 2000, intentan fijar cuantas componentes sean posibles. Para ello, partiendo de la formulación en cliques maximales, siguen distintas estrategias, que se detallan a continuación.

Restricciones GUB

La formulación en cliques se puede reescribir añadiendo variables de holgura $s \in \{0, 1\}^{|\mathcal{C}|}$ en las restricciones. Las igualdades que resultan se conocen con el término *generalised upper bound (GUB) constraints*. Sea

$$P_C^{GUB} = \left\{ (\mathbf{x}, \mathbf{s}) \in \mathbb{R}^{|V|} \times \mathbb{R}^{|\mathcal{C}|} : \sum_{v \in C} x_v + s_C = 1 \forall C \in \mathcal{C}, \mathbf{x} \geq \mathbf{0}, \mathbf{s} \geq \mathbf{0} \right\}.$$

La ventaja de tener una formulación con restricciones GUB es que podemos definir un esquema de ramificación específico, conocido como ramificación en GUBs. Consideremos una iteración i del algoritmo de ramificación y acotación para la que la relajación lineal $\bar{\Pi}_i$ que se resuelve es factible. Esta relajación es el problema

$$\begin{aligned} (\bar{\Pi}_i) \quad & \text{máx} \quad z(\mathbf{x}) = \mathbf{c}^T \mathbf{x} \\ & \text{s.a} \quad A\mathbf{x} = \mathbf{b} \\ & \quad \ell^i \leq \mathbf{x} \leq \mathbf{u}^i \end{aligned}$$

donde $\mathbf{x} = (\mathbf{x}_V, \mathbf{x}_C)$ se refiere tanto a las variables que representan los nodos de $G = (V, E)$ como a las variables de holgura $s \in \{0, 1\}^{|\mathcal{C}|}$. El vector de costes es $\mathbf{c} = (\mathbf{c}_V, \mathbf{c}_C) = (\mathbf{1}_V, \mathbf{0}_C)$, A es una matriz binaria de tamaño $|\mathcal{C}| \times |V| + |\mathcal{C}|$ y $\mathbf{b} = \mathbf{1}_C$. Para cada clique $C \in \mathcal{C}$, la variable s_C se puede interpretar como una variable correspondiente a un nodo extra en G que tiene coeficiente cero en la función objetivo y que es adyacente a todos los nodos de C . Denotaremos esta extensión del grafo original como $G' = (V', E')$.

En la iteración i , consideremos una solución óptima al problema $(\bar{\Pi}_i)$, $(\bar{\mathbf{x}}_V^i, \bar{\mathbf{x}}_C^i)$, con base asociada B . Los costes reducidos son $c_j^\pi = c_j - c_B B^{-1} a_{.j}$, para cada $j = 1, \dots, |V'|$. Sea

$z^i = z(\bar{\mathbf{x}}^i)$ y $L, U \subseteq V'$ conjuntos índices tales que $\mathbf{c}_L^\pi < 0$ y $\mathbf{c}_U^\pi > 0$. Denotemos con z^* el valor objetivo correspondiente a la mejor solución entera encontrada hasta el momento. Se definen las siguientes cotas $\tilde{\mathbf{l}}^i$ y $\tilde{\mathbf{u}}^i \in \{0, 1\}^{|V'|}$:

$$\tilde{l}_j^i = \begin{cases} \max \left\{ 0, 1 + \left\lceil \frac{z^* - z^i}{\min\{-\tilde{c}_k^\pi | k \in N(j)\}} \right\rceil \right\} & \text{si } j \in U, \\ 0 & \text{en otro caso} \end{cases}$$

y

$$\tilde{u}_j^i = \begin{cases} \min \left\{ 1, \left\lfloor \frac{z^* - z^i}{\tilde{c}_j^\pi} \right\rfloor \right\} & \text{si } j \in L, \\ 1 & \text{en otro caso,} \end{cases}$$

donde $\tilde{c}_j^\pi = c_j^\pi - \sum_{k \in N(j) \cap U} c_k^\pi$.

Proposición 2.3.1 (Strijk et al., 2000). *Todos los vectores enteros \mathbf{x}^{int} factibles del problema (Π_i) con $z(\mathbf{x}^{int}) \geq z^*$ satisfacen $\tilde{\mathbf{l}}^i \leq \mathbf{x}^{int} \leq \tilde{\mathbf{u}}^i$.*

Demostración. Sean $\bar{\mathbf{x}}^i, z^i, L, U$ y z^* los definidos previamente en la construcción de $\tilde{\mathbf{l}}^i$ y $\tilde{\mathbf{u}}^i$. Supongamos que \mathbf{x}^{int} es factible del problema (Π_i) con $z(\mathbf{x}^{int}) \geq z^*$. Podemos escribir los valores objetivo en función de los costes reducidos como:

$$z(\mathbf{x}^{int}) = \mathbf{c}_B B^{-1} \mathbf{b} + (\mathbf{c}_L^\pi)^T \mathbf{x}_L^{int} + (\mathbf{c}_U^\pi)^T \mathbf{x}_U^{int},$$

$$z(\bar{\mathbf{x}}^i) = \mathbf{c}_B B^{-1} \mathbf{b} + (\mathbf{c}_L^\pi)^T \mathbf{l}_L^i + (\mathbf{c}_U^\pi)^T \mathbf{c}_U^i.$$

Sea $j \in L$ cualquiera. Si $x_j^{int} = 0$ entonces se cumple que $x_j^{int} \leq \tilde{u}_j^i$ porque \tilde{u}_j^i es positivo. Supongamos entonces que $x_j^{int} = 1$. Entonces, por las restricciones GUB se cumple que $x_{N(j)}^{int} = \mathbf{0}$. Por lo tanto,

$$\begin{aligned} z^* - z^i &\leq z(\mathbf{x}^{int}) - z(\bar{\mathbf{x}}^i) \\ &= (\mathbf{c}_L^\pi)^T (\mathbf{x}_L^{int} - \mathbf{l}_L^i) + (\mathbf{c}_U^\pi)^T (\mathbf{x}_U^{int} - \mathbf{c}_U^i) \\ &= (\mathbf{c}_L^\pi)^T \mathbf{x}_L^{int} + (\mathbf{c}_U^\pi)^T (\mathbf{x}_U^{int} - \mathbf{1}) \\ &\leq c_j^\pi x_j^{int} + (\mathbf{c}_{U \cap N(j)}^\pi)^T (\mathbf{x}_{U \cap N(j)}^{int} - \mathbf{1}) \\ &= c_j^\pi - \sum_{k \in U \cap N(j)} c_k^\pi. \end{aligned}$$

La segunda desigualdad que aparece en la cadena anterior se debe a que, por la optimalidad de $\bar{\mathbf{x}}^i$, $\mathbf{c}_L^\pi < 0$ y $\mathbf{c}_U^\pi > 0$, y por la factibilidad de \mathbf{x}^{int} , $\mathbf{x}_L^{int} \geq 0$ y $\mathbf{x}_U^{int} \leq 1$. Por lo tanto, $(\mathbf{c}_L^\pi)^T \mathbf{x}_L^{int} + (\mathbf{c}_U^\pi)^T (\mathbf{x}_U^{int} - \mathbf{1})$ es una suma de términos negativos y la suma de un subconjunto de estos términos será mayor que la suma total.

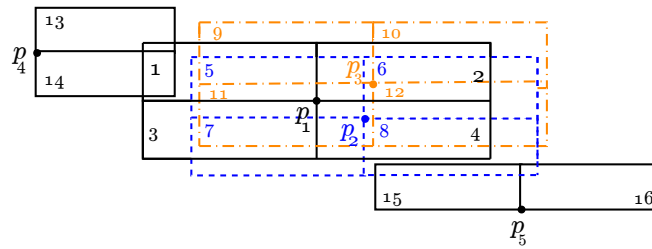
Como $c_j^\pi - \sum_{k \in U \cap N(j)} c_k^\pi < 0$ (pues $j \in L$ y $k \in U$) tenemos que

$$\frac{z^* - z^i}{c_j^\pi - \sum_{k \in U \cap N(j)} c_k^\pi} \geq 1.$$

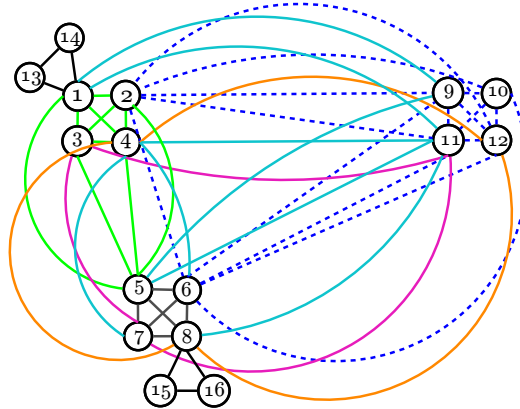
Así, $x_j^{int} = 1 \leq \tilde{u}_j^i$. Hemos demostrado que en ambos casos, $x_j^{int} = 0$ y $x_j^{int} = 1$, se cumple que $x_j^{int} \leq \tilde{u}_j^i$. Como j se eligió de forma arbitraria tenemos que $\mathbf{x}^{int} \leq \tilde{\mathbf{u}}^i$.

De manera similar se prueba que $\tilde{\ell}^i \leq \mathbf{x}^{int}$, partiendo de $j \in U$, asumiendo que $x_j^{int} = 0$ y observando que esto implica por las restricciones GUB que $x_k^{int} = 1$ para algún $k \in N(j)$, lo que a su vez supone $\mathbf{x}_{N(k)}^{int} = 0$. \square

El Ejemplo 2.3.1 ilustra la fijación de variables usando las cotas anteriormente descritas.



(a) Puntos y regiones de etiquetado



(b) Grafo intersección asociado

Figura 2.2: Instancia del problema de etiquetado y su grafo intersección para el Ejemplo 2.3.1

Ejemplo 2.3.1. Supongamos que disponemos de una instancia del problema MNE formada por cinco puntos, $\mathcal{P} = \{p_1, \dots, p_5\}$, para cada uno de los cuales hay una cantidad finita de opciones de etiquetado, como muestra la Figura 2.2a. Como $|L_i| = 4$ para $i = 1, 2, 3$ y $|L_i| = 2$ para $i = 4, 5$, en total hay 16 ubicaciones posibles de las etiquetas. De esta forma, el grafo

intersección asociado, que se muestra en la Figura 2.2b, tendrá 16 nodos. La siguiente se corresponde con una formulación en cliques maximales del problema. La numeración de las variables x_i , $i = 1, \dots, 16$ es la misma que la que aparece en la Figura 2.2a para las posibles ubicaciones de las etiquetas.

$$\begin{aligned}
& \text{máx} && \sum_{i=1}^{20} x_i \\
& \text{s.a.} && x_2 + x_6 + x_9 + x_{10} + x_{11} + x_{12} \leq 1 \\
& && x_5 + x_1 + x_2 + x_3 + x_4 \leq 1 \\
& && x_5 + x_6 + x_7 + x_8 \leq 1 \\
& && x_1 + x_{13} + x_{14} \leq 1 && x_8 + x_{15} + x_{16} \leq 1 \\
& && x_3 + x_7 + x_{11} \leq 1 && x_4 + x_8 + x_{12} \leq 1 \\
& && x_1 + x_9 \leq 1 && x_1 + x_{11} \leq 1 \\
& && x_4 + x_6 \leq 1 && x_4 + x_7 \leq 1 \\
& && x_4 + x_{11} \leq 1 && x_5 + x_9 \leq 1 \\
& && x_5 + x_{11} \leq 1 && x_8 + x_{11} \leq 1 \\
& && x_i \in \{0, 1\} && \forall i = 1, \dots, 16.
\end{aligned}$$

La Figura 2.2b muestra con distintos colores cada una de las cliques maximales del grafo intersección, salvo las cliques de tamaño dos, para las que se ha usado un mismo color, el azul claro. Antes de comenzar el algoritmo de ramificación, es necesario transformar las restricciones de clique en restricciones GUB, añadiendo en total 15 variables de holgura, s_j , $j = 1, \dots, 15$. El índice j recorre las restricciones de la formulación anterior en primer lugar de izquierda a derecha y en segundo de arriba a abajo. La solución de la relajación lineal del problema resultante es: $x_3 = x_4 = x_5 = 0.333$, $x_7 = s_7 = s_{10} = s_{12} = s_{13} = s_{14} = 0.667$, $x_{10} = x_{13} = x_{15} = s_8 = s_9 = s_{15} = 1$, con valor objetivo 4.667. Una vez obtenida esta solución fraccional, comienza el proceso de fijación de variables. Los costes reducidos de las variables no básicas son $c_{x_1}^\pi = c_{x_2}^\pi = c_{s_2}^\pi = -0.667$, $c_{x_6}^\pi = c_{x_8}^\pi = c_{x_{11}}^\pi = c_{s_3}^\pi = c_{s_6}^\pi = c_{s_{11}}^\pi = -0.333$, $c_{s_1}^\pi = c_{s_4}^\pi = c_{s_5}^\pi = -1$ (y el resto cero). Como todos los costes reducidos son negativos, $U = \emptyset$, y $\tilde{c}_j^\pi = c_j^\pi$ para todo índice j no básico. Como el algoritmo aún no ha encontrado ninguna solución entera, tomamos $z^* = 0$ (una solución factible es no etiquetar ningún punto). Dado que los costes reducidos son negativos, se tiene que $\tilde{l}_j = 0$ y $\tilde{u}_j = \min\{1, \lfloor -4.667/c_j^\pi \rfloor\} = 1$ para cada índice no básico j . En este caso no se fijaría ninguna variable. Para continuar,

se tomaría una de las variables básicas con valor no entero y se ramificaría a partir de ella, formando dos subproblemas, uno en el que la variable estuviera fijada a 0 y otro en el que se fijara a 1. Para ilustrar el proceso de fijación de variables, supongamos que en lugar de $z^* = 0$ hemos encontrado una solución factible de manera que $z^* = 4$ (por ejemplo, $x_3 = x_{10} = x_{13} = x_{16} = 1$). Observamos que el problema no se descarta puesto que $z = 4,667 > z^*$. En este caso $\tilde{u}_{s_1} = \tilde{u}_{s_4} = \tilde{u}_{s_5} = \min\{1, \lfloor (4 - 4.667) / -1 \rfloor\} = 0$, $\tilde{u}_j = 1$ para el resto de variables. Esta vez se consigue estrechar las cotas para las variables s_1 , s_4 y s_5 . Cuando ramifiquemos en una de las variables que toma valor fraccional, fijaremos $s_1 = s_4 = s_5 = 0$ en los subproblemas resultantes.

Implicaciones lógicas

En cada nodo del árbol de ramificación, se resuelve un problema de la forma (Π_i) y se emplea la Proposición 2.3.1 para establecer nuevas cotas y así fijar las variables que sea posible. Una vez hecho esto, se aprovechan ciertas propiedades de la formulación para fijar más variables. En concreto, se utilizan las dos implicaciones siguientes.

Proposición 2.3.2. *Sea $v \in V$ un nodo de G . Si x_v se fija a uno, entonces x_w puede fijarse a cero para todo $w \in N(v)$ sin cambiar el valor óptimo del problema resultante.*

Proposición 2.3.3. *Sea $v \in V$ un nodo de G . Si x_w se fija a cero para todo $w \in N(v)$, entonces x_v puede fijarse a uno sin cambiar el valor óptimo del problema resultante.*

Recursión y sustitución

De nuevo, supongamos que nos encontramos en una iteración del algoritmo de ramificación y corte. Asumamos además que se han añadido hiperplanos de corte, establecido nuevas cotas y aplicado las proposiciones 2.3.2 y 2.3.3. Como resultado, se tendrán $J_0, J_1 \subseteq V$ los conjuntos de nodos cuyas variables correspondientes han sido fijadas a cero y uno respectivamente, y $F = V \setminus (J_0 \cup J_1)$ el conjunto de nodos cuyas variables asociadas son libres. Sea $\{F_1, \dots, F_k\}$ una partición de F de manera que $G[F_i]$ es una componente conexa de $G[F]$. La siguiente proposición establece la solución óptima entera en el conjunto

$$P_E \cap \{\mathbf{x} \in \mathbb{R}^{|V|} \mid \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}$$

combinando soluciones óptimas enteras en los subconjuntos F_i .

Proposición 2.3.4. *En las condiciones anteriores, sea $i \in \{1, \dots, k\}$ y sea $I \subseteq F_i$ un empaquetamiento de cardinal máximo en $G[F_i]$. Entonces, alguno de los empaquetamientos I^* en G de cardinal máximo bajo las restricciones $J_0 \cap I^* = \emptyset$ y $J_1 \subseteq I^*$ cumple que $I \subseteq I^*$.*

Demostración. Sea I_1 un empaquetamiento de cardinal máximo en G cumpliendo que $J_0 \cap I_1 = \emptyset$ y $J_1 \subseteq I_1$. Sea $I_2 = I_1 \setminus F_i$ e $I^* = I_2 \cup I$. Vamos a ver que I^* es un empaquetamiento de cardinal máximo cumpliendo las condiciones del enunciado.

En primer lugar, se cumple que $J_0 \cap I^* = \emptyset$ y $J_1 \subseteq I^*$ por cumplir I_1 las propiedades y por la construcción de I^* a partir de I_1 . Por otro lado, es evidente que $I \subseteq I^*$. Con el fin de completar la prueba, sólo queda ver que I^* es un empaquetamiento de cardinal máximo. Para cualquier $v \in I$ y cualquier nodo adyacente w , $(v, w) \in E$, se cumple que $w \in F_i \cup J_0$ (en particular $w \notin I_2$). Se sigue que I^* es un empaquetamiento en G . Veamos que además es de cardinal máximo con las propiedades requeridas. Como $I_1 \cap F_i$ es un empaquetamiento de nodos de $G[F_i]$, tenemos que

$$|I_2| = |I_1 \setminus F_i| = |I_1| - |I_1 \cap F_i| \geq |I_1| - |I|.$$

Como $I \cap I_2 = \emptyset$, esto implica que

$$|I^*| = |I_2 \cup I| = |I_2| + |I| \geq |I_1| - |I| + |I| \geq |I_1|,$$

y como I_1 era de cardinal máximo también lo es I^* . □

Como último paso para fijar variables, en Strijk et al., 2000 los autores llevan a cabo el siguiente procedimiento. Consideran F_1 la mayor componente conexa en $G[F]$, es decir, $|F_1| \geq |F_i|$ para cualquier $i \in \{2, \dots, k\}$. Para cada $i \in \{2, \dots, k\}$, se calcula un empaquetamiento de cardinal máximo I_i^* en $G[F_i]$ y se fijan las variables de I_i^* a uno y las de $F_i \setminus I_i^*$ a cero. Las variables correspondientes a F_1 quedan libres. Debido a la Proposición 2.3.4, esto no reduce el valor objetivo de la solución óptima del problema del empaquetamiento de cardinal máximo.

2.3.3. Desigualdades válidas y algoritmos de separación

Hasta ahora nos hemos centrado en uno de los dos aspectos novedosos del algoritmo de ramificación y corte de Verweij, la fijación de variables. Esta subsección está dedicada al estudio del segundo aspecto original del algoritmo, que es el procedimiento de separación. Este procedimiento tiene lugar al comienzo de cada etapa del algoritmo de ramificación y consiste en reforzar la relajación lineal del problema actual, añadiendo desigualdades válidas que sean

violadas por la solución óptima fraccional encontrada. Normalmente, se dispone de una serie de familias de desigualdades conocidas del problema y el procedimiento de separación establece las estrategias para seleccionar aquellas que deben ser añadidas. Cuantas más restricciones sean incluidas se obtendrá un problema más reforzado y como contrapartida más costoso de resolver.

En Verweij, 2000, los autores desarrollan un algoritmo de separación que trabaja con una clase de desigualdades conocida para el problema del empaquetamiento de cardinal máximo, las desigualdades asociadas a agujeros impares. El procedimiento consiste en identificar cuáles de estas restricciones están cerca de ser violadas por la solución. A continuación, se refuerzan las restricciones identificadas, añadiendo variables en el lado izquierdo de las desigualdades mediante un proceso conocido como levantamiento de facetas. Después, se comprueba si las restricciones reforzadas efectivamente son violadas y, en tal caso, se añaden al problema. En esta subsección se describe detalladamente este algoritmo de separación. En él se distinguen dos partes bien diferenciadas. Primero se identifica un agujero impar en el grafo intersección del problema de etiquetado de cardinal máximo. Después se refuerza la desigualdad que viene asociada a este agujero. A continuación se presentan los conceptos que intervienen en este algoritmo de separación, como los de agujero impar, levantamiento o faceta. Seguidamente se describen en detalle las dos partes que constituyen el algoritmo.

Conceptos previos

A continuación damos una serie de definiciones que se refieren a algunos elementos de un poliedro. En nuestro ámbito de trabajo, este poliedro estará asociado a la región factible de un problema de optimización. La Figura 2.3 sirve de ilustración de estos conceptos.

Definición 2.3.1. *La dimensión de un poliedro $P \subseteq \mathbb{R}^n$ es k si el máximo número de puntos afínmente independientes en P es $k+1$ (recordemos que un conjunto de puntos $\{a_0, a_1, \dots, a_n\} \subseteq \mathbb{R}^n$ se dice afínmente independiente si los vectores $\{\overrightarrow{a_0a_1}, \dots, \overrightarrow{a_0a_n}\}$ son linealmente independientes). Un poliedro $P \subseteq \mathbb{R}^n$ es de dimensión completa si $\dim(P) = n$.*

Definición 2.3.2. *Una desigualdad $\pi x \leq \pi_0$, con $\pi \in \mathbb{R}^n$, $\pi_0 \in \mathbb{R}$, se llama desigualdad válida del poliedro $P \subseteq \mathbb{R}^n$ si es satisfecha por todos los puntos de P .*

Definición 2.3.3. *Si $\pi x \leq \pi_0$ es una desigualdad válida del poliedro P , al conjunto $F = \{x \in P : \pi x = \pi_0\}$ se le llama cara de P . La cara F es propia si $F \neq \emptyset$ y $F \neq P$.*

Un agujero impar en un grafo $G = (V, E)$ es un ciclo $H \subseteq V$ sin cuerdas con cardinal $|H|$ impar. Asociada a un agujero impar, puede definirse la siguiente desigualdad:

$$x(H) \leq \frac{|H| - 1}{2},$$

donde $x(H) = \sum_{v \in H} x_v$. Esta desigualdad es válida para P_{conv} y constituye una faceta del poliedro $P_{conv} \cap \{\mathbf{x} | \mathbf{x}_{V \setminus H} = \mathbf{0}\}$ (Padberg, 1973). Es decir, es faceta del poliedro asociado al problema de empaquetamiento en el subgrafo inducido por H . Dado un ciclo impar, una desigualdad levantada asociada al ciclo impar es de la forma:

$$x(H) + \sum_{v \in V \setminus H} \pi_v x_v \leq \frac{|H| - 1}{2},$$

para un vector adecuado de coeficientes $\boldsymbol{\pi} \in \mathbb{R}^{|V \setminus H|}$. Estos coeficientes se calculan aplicando sucesivamente el Teorema 2.3.5, de manera que la desigualdad anterior que resulta de este proceso de levantamiento será una faceta del poliedro P_{conv} .

Identificación de un agujero impar casi violado

El procedimiento que identifica un agujero impar utilizado por Verweij se basa en una construcción que fue propuesta originalmente por Grötschel et al., 1988. Este procedimiento tiene lugar en varios pasos, que se detallan a continuación.

Partimos de un grafo $G = (V, E)$ asociado al problema del empaquetamiento de cardinal máximo. El primer paso consiste en construir un grafo bipartito $\tilde{G} = ((\tilde{V}^1, \tilde{V}^2), \tilde{E})$ y en asociar unos vectores de coste $\mathbf{c} \in [0, 1]^{|E|}$ y $\tilde{\mathbf{c}} \in [0, 1]^{|\tilde{E}|}$ a las aristas de G y \tilde{G} respectivamente. Estos nuevos elementos tienen carácter auxiliar y servirán de apoyo para finalmente obtener el agujero impar en G buscado. La construcción se lleva a cabo como sigue. Cada nodo $v \in V$ se divide en dos nodos $v^1 \in \tilde{V}^1$ y $v^2 \in \tilde{V}^2$. Para cada arista $(u, v) \in E$ se añaden a \tilde{E} las aristas (u^1, v^2) y (u^2, v^1) . Finalmente, se establecen los costes

$$c_{(u,v)} = \tilde{c}_{(u^1,v^2)} = \tilde{c}_{(u^2,v^1)} = 1 - x_u - x_v \quad \forall (u, v) \in E.$$

Estos costes están entre 0 y 1 debido a las restricciones asociadas al problema de empaquetamiento sobre G . La Figura 2.4 muestra un ejemplo de esta construcción.

Observamos que, por la forma de construir el grafo \tilde{G} , un camino de $u^1 \in \tilde{V}^1$ a $v^2 \in \tilde{V}^2$

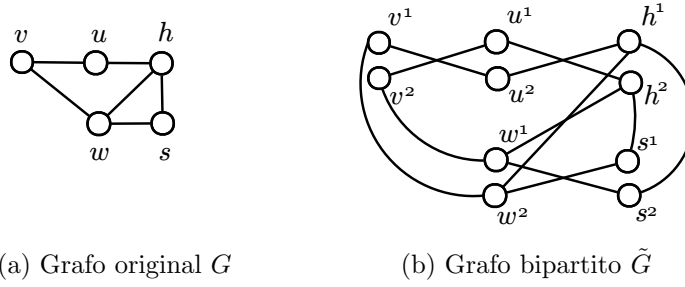


Figura 2.4: Ejemplo de la construcción propuesta por Grötschel et al., 1988

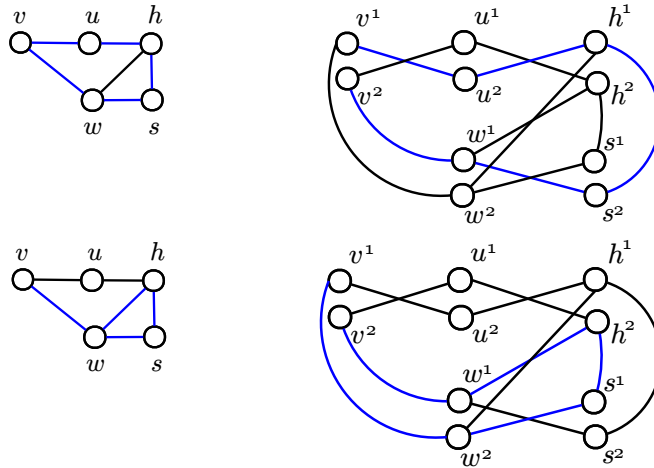


Figura 2.5: Correspondencia entre paseos y caminos más cortos en G y \tilde{G}

se corresponde con un paseo de longitud impar en G desde u hasta v . Además, un camino de $v^1 \in \tilde{V}^1$ a $v^2 \in \tilde{V}^2$ en \tilde{G} se corresponde con un paseo cerrado de longitud impar en G conteniendo a v . Un par de ejemplos de esta última correspondencia pueden verse en la Figura 2.5. Estos ejemplos se ilustran sobre el mismo grafo mostrado en la Figura 2.4. Respecto al par de grafos de la parte superior de la Figura 2.5, el camino del grafo \tilde{G} (a la derecha), $(v^1, u^2, h^1, s^2, w^1, v^2)$, se corresponde con el paseo cerrado de longitud impar (v, u, h, s, w, v) en G (a la izquierda). En la parte inferior de la figura, el camino $(v^1, w^2, s^1, h^2, w^1, v^2)$ del grafo \tilde{G} se corresponde en este caso con el paseo cerrado de longitud impar (v, w, s, h, w, v) en G .

Fijémonos ahora en los costes de los caminos y paseos en \tilde{G} y G respectivamente. Nos referiremos al coste del camino o paseo como la suma de los costes de las aristas que forman parte de este, es decir, \tilde{c} y c respectivamente. Un camino más corto (de coste mínimo) de $v^1 \in \tilde{V}^1$ a $v^2 \in \tilde{V}^2$ en \tilde{G} se corresponde con un paseo más corto cerrado de longitud impar en G conteniendo a v . El motivo de esta correspondencia es que el número de aristas coincide para ambos y también los costes (pues así han sido definidos). Por otro lado, es fácil comprobar que

un agujero impar H en G define una desigualdad violada si

$$\sum_{\substack{u,v \in H \\ (u,v) \in E}} c_{(u,v)} < 1. \quad (2.3)$$

En efecto, la desigualdad violada asociada al agujero es:

$$\sum_{v \in H} x_v > \frac{|H| - 1}{2},$$

$$|H| - 2 \sum_{v \in H} x_v < 1,$$

$$\sum_{v \in H} (1 - 2x_v) < 1.$$

Al mismo tiempo, como cada nodo de H es adyacente exactamente a dos nodos de este agujero tenemos que

$$\sum_{\substack{u,v \in H \\ (u,v) \in E}} c_{(u,v)} = \sum_{\substack{u,v \in H \\ (u,v) \in E}} (1 - x_u - x_v) = \sum_{v \in H} (1 - 2x_v).$$

El segundo paso de la construcción llevada a cabo por Verweij es, fijado $v \in V$, encontrar el camino más corto en \tilde{G} desde v^1 hasta v^2 y obtener así el correspondiente paseo más corto en G . Modificando este último, se obtiene el agujero impar buscado. La razón por la que Verweij se interesa por un camino más corto en \tilde{G} es que es más probable que al final del proceso obtengamos una desigualdad violada que si partimos de un camino de coste mayor, debido a la condición (2.3).

Supongamos entonces que, fijado un nodo $v \in V$, partimos de un paseo cerrado en G que lo contiene:

$$C := (v = v_0, v_1, v_2, \dots, v_{k-1}, v_k = v) \quad (2.4)$$

con k impar, de coste mínimo, obtenido a partir de un camino en \tilde{G} desde v^1 hasta v^2 de coste mínimo. Además, en caso de empate elegimos C minimal con respecto a $|C|$. Algunos nodos pueden aparecer en C más de una vez y además puede haber cuerdas uniendo nodos del paseo.

Como tercer paso para obtener un agujero impar a partir de C se procede como sigue. Consideramos $j \in \{2, \dots, k-1\}$ el menor índice de forma que existe una arista $(v_i, v_j) \in E$ para algún $i \in \{0, \dots, j-2\}$ (este par de índices existe porque $(v_0, v_{k-1}) \in E$). Sea $i \in \{0, \dots, j-2\}$

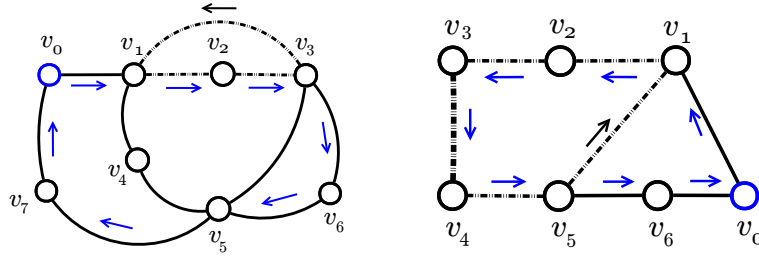


Figura 2.6: Construcción de un agujero impar

el mayor índice tal que $(v_i, v_j) \in E$. Se define

$$H := (v_i, v_{i+1}, \dots, v_{j-1}, v_j, v_i).$$

La construcción de H descrita se ilustra en la Figura 2.6 a través de dos ejemplos independientes. En cada uno de ellos se muestra un paseo en el grafo G . El nodo inicial del paseo se destaca en color azul y la dirección de las aristas del paseo se indica con flechas azules. Las aristas del agujero extraído del paseo aparecen con trazo discontinuo. En el ejemplo de la izquierda de la figura, el paseo es $C = (v_0, v_1, v_2, v_3, v_6, v_5, v_7, v_0)$, $j = 3$ e $i = 1$, por lo que el agujero resultante es $H = (v_1, v_2, v_3, v_1)$. En el ejemplo de la derecha, el paseo es $C = (v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_0)$, $j = 5$ e $i = 1$, por lo que el agujero resultante es $H = (v_1, v_2, v_3, v_4, v_5, v_1)$. Supongamos que suprimimos la arista (v_1, v_5) del grafo de la derecha y en su lugar añadimos la arista (v_2, v_5) . Aplicando de nuevo el procedimiento para generar un agujero obtendríamos $H = (v_2, v_3, v_4, v_5, v_2)$, que es par. En el ejemplo no hemos tenido en cuenta los costes de las aristas y es precisamente esta la razón por la que a priori parece que el procedimiento puede producir un agujero par. La siguiente proposición muestra que esto no ocurre cuando C es un paseo impar de coste mínimo como en (2.4).

Proposición 2.3.6 (Verweij, 2000). *Sea C un paseo como en (2.4), entonces el agujero H que se obtiene a partir de él por el procedimiento anterior es un agujero impar.*

Demostración. Como $(v^1, v^2) \notin \tilde{E}$, tenemos que $|H| \geq 3$ (H no es trivial). Claramente, H es un ciclo en G . Por la elección de i y j , H no contiene cuerdas, por lo que es un agujero. Es necesario entonces probar que $|H| = j - i + 1$ es impar. Por reducción al absurdo, supongamos que $|H|$ es par. Entonces

$$C' := (v = v_0, \dots, v_{i-1}, v_i, v_j, v_{j+1}, \dots, v_k = v)$$

es un paseo impar cerrado en G que contiene a v . Además, si calculamos la suma de los costes de las aristas $e_p = (v_{p-1}, v_p)$ que han quedado fuera de C' , tenemos:

$$c(\{e_{i+1}, \dots, e_j\}) = (j - i) - \left(2 \sum_{p=i+1}^{j-1} x_{v_p}\right) - x_{v_i} - x_{v_j}.$$

Por las restricciones de empaquetamiento $x_{v_p} + x_{v_{p+1}} \leq 1$, lo cual implica que

$$\sum_{p=i+1}^{j-1} x_{v_p} \leq \frac{j - i - 1}{2}.$$

Por lo tanto,

$$c(\{e_{i+1}, \dots, e_j\}) \geq (j - i) - (j - i - 1) - x_{v_i} - x_{v_j} = c_{(v_i, v_j)}.$$

En definitiva, al coste total de C le hemos quitado una cantidad mayor que $c_{(v_i, v_j)}$ y le hemos añadido exactamente $c_{(v_i, v_j)}$, por lo que el coste de C' que resulta es menor o igual que el de C . Si el coste de C' es menor obtenemos una contradicción, pues C era un paseo cerrado impar conteniendo a v de coste mínimo. Por tanto, la suma de los costes de las aristas de C y C' coinciden. Sin embargo, el cardinal de C' es menor, lo que contradice la elección de C . Como suponiendo que $|H|$ es par llegamos a contradicción, $|H|$ debe ser impar. \square

A la hora de llevar a la práctica el procedimiento descrito para encontrar un agujero impar, es necesario encontrar un camino más corto en el grafo \tilde{G} . Como los costes son no negativos, este camino se pueden hallar aplicando el algoritmo de Dijkstra, 1959. El proceso seguido para extraer el agujero H es totalmente automatizable y se puede implementar siguiendo el siguiente esquema:

0. Partir de un grafo $G = (V, E)$ y una solución $\mathbf{x}^{|V|}$ de la relajación lineal del problema de empaquetamiento en G .
1. Construir el grafo bipartito $\tilde{G} = ((\tilde{V}^1, \tilde{V}^2), \tilde{E})$ y asociar a sus aristas los costes \tilde{c} .
2. Elegir $v \in V$ y aplicar el algoritmo de Dijkstra para encontrar \tilde{C} , el camino más corto en \tilde{G} desde v^1 hasta v^2 .
3. Traducir \tilde{C} a un paseo $C = (v = v_0, v_1, \dots, v_{k-1}, v_k = v_0)$ en G .

4. Calcular:

$$j = \min\{r \in \{2, \dots, k-1\} : \exists i \in \{0, \dots, j-2\}, (v_i, v_r) \in E\},$$

$$i = \max\{s \in \{0, \dots, j-2\} : (v_s, v_j) \in E\},$$

$$H = (v_i, v_{i+1}, \dots, v_{j-1}, v_j, v_i).$$

Una vez hallado el agujero impar H , se descartan los costes \tilde{c} así como el grafo auxiliar \tilde{G} , que no intervendrán en el resto del algoritmo de separación. La parte restante de este algoritmo consiste en reforzar la desigualdad asociada a H , para convertirla en una faceta del problema del empaquetamiento sobre G . Esto se consigue mediante el proceso de levantamiento dado en el Teorema 2.3.5. A continuación se describe el procedimiento utilizado por Verweij, 2000 para levantar el agujero impar encontrado.

Levantamiento de un agujero impar

Supongamos que H es un agujero en $G = (V, E)$ obtenido por el procedimiento anterior. Como ya se ha mencionado, este agujero da lugar a una faceta del problema de empaquetamiento asociado al subgrafo inducido $G[H]$:

$$\sum_{v \in H} x_v \leq \frac{|H| - 1}{2}.$$

Para transformarla en una faceta del problema original, es necesario considerar los nodos de $V \setminus H$ en cierto orden y aplicar levantamientos sucesivos, según se indica en el Teorema 2.3.5. Supongamos que en una de estas iteraciones tenemos $V' \subseteq V$ el conjunto de nodos de H junto con aquellos nodos a los que ya se les ha aplicado el levantamiento. Tendremos entonces la faceta de $G[V']$:

$$\sum_{v \in H} x_v + \sum_{v \in V' \setminus H} \pi_v x_v \leq \frac{|H| - 1}{2},$$

para cierto vector de coeficientes π . Sea $u \in V \setminus V'$ el siguiente nodo que se va a levantar. Según el Teorema 2.3.5, el coeficiente de x_u se obtiene calculando el empaquetamiento de mayor peso en $G[V' \setminus N(u)]$. Cuando hablamos del peso del empaquetamiento nos referimos a la suma $\sum_{v \in V' \setminus N(u)} \pi_v x_v$ (donde $x_v = 1$ si v pertenece al empaquetamiento y $x_v = 0$ en otro caso).

Por lo tanto, el levantamiento de la desigualdad asociada a H se traduce en la resolución

de $|V \setminus H|$ problemas de empaquetamiento. Nemhauser and Sigismondi, 1992 observaron que $\pi_v = 0$ para $v \in V \setminus H$ si $|N(v) \cap H| \leq 2$. Esto implica que, en la práctica, el número de problemas de levantamiento que hay que resolver es relativamente pequeño. Para calcular los coeficientes del levantamiento, π , Verweij, 2000 utiliza el concepto de descomposición de caminos, con el que ya se había trabajado en Bodlaender, 1997 y Fluiter, 1997.

Definición 2.3.5. Una descomposición de caminos de un grafo $G = (V, E)$ es una secuencia $(S_i)_{i=1}^q$ de conjuntos de nodos satisfaciendo:

$$\bigcup_{i=1}^q S_i = V, \quad (2.5)$$

$$\forall (u, v) \in E, \exists i \in \{1, \dots, q\} \{u, v\} \subseteq S_i, \quad (2.6)$$

$$\forall i, j, k, 1 \leq i < j < k \leq q, S_i \cap S_k \subseteq S_j. \quad (2.7)$$

Diremos que la descomposición es normalizada si $\forall i \in \{1, \dots, q-1\}$ S_i y S_{i+1} difieren en solo un nodo, es decir, $S_{i+1} = S_i \cup \{v\}$ ($v \notin S_i$) o $S_{i+1} = S_i \setminus \{v\}$ ($v \in S_i$) para algún $v \in V$, y además $S_1 = S_q = \emptyset$. En este caso se cumple que $q = 2|V| + 1$.

Dado el agujero impar $H = (v_0, v_1, \dots, v_n = v_0)$, una descomposición de caminos del grafo $G[H]$ viene dada por:

$$S_i = \begin{cases} \emptyset & \text{si } i = 1 \text{ o } i = 2|H| + 1, \\ \{v_0\} & \text{si } i = 2 \text{ o } i = 2|H|, \\ \{v_0, v_k\} & \text{si } 2 < i < 2|H| \text{ e } i = 2k + 1, \\ \{v_0, v_k, v_{k+1}\} & \text{si } 2 < i < 2|H| \text{ e } i = 2(k + 1). \end{cases} \quad (2.8)$$

Ejemplo 2.3.2. Supongamos que $H = (v_0, v_1, v_2, v_3, v_4)$ es el ciclo impar de cinco nodos. Entonces, aplicando la descomposición de caminos descrita arriba obtenemos los siguientes conjuntos:

$$\begin{aligned} S_1 &= \emptyset, & S_5 &= \{v_0, v_2\}, & S_9 &= \{v_0, v_4\}, \\ S_2 &= \{v_0\}, & S_6 &= \{v_0, v_2, v_3\}, & S_{10} &= \{v_0\}, \\ S_3 &= \{v_0, v_1\}, & S_7 &= \{v_0, v_3\}, & S_{11} &= \emptyset. \\ S_4 &= \{v_0, v_1, v_2\}, & S_8 &= \{v_0, v_3, v_4\}, \end{aligned}$$

Se puede comprobar fácilmente que estos conjuntos cumplen las tres propiedades (2.5), (2.6) y (2.7).

En cierto instante durante el desarrollo del procedimiento de levantamiento, tendremos

$V' \subseteq V$ el conjunto de nodos que o bien están en H o bien les ha sido ya aplicado el levantamiento y $u \in V \setminus V'$ el siguiente nodo que se va a levantar. El algoritmo de levantamiento mantiene en todo momento una descomposición de caminos para el grafo $G[V']$. Esta descomposición se irá actualizando y se utilizará para calcular en cada paso el empaquetamiento de mayor peso en $G[V' \setminus N(u)]$. Este empaquetamiento determinará el coeficiente de x_u en la desigualdad levantada.

Si $(S_i)_{i=1}^{2|V'|+1}$ es una descomposición de caminos para el grafo $G[V']$, una descomposición de caminos para $G[V' \setminus N(u)]$ se puede obtener a partir de la anterior eliminando los nodos de $N(u)$ de todos los conjuntos S_i ($i = 1, \dots, 2|V'| + 1$) y eliminando los conjuntos consecutivos que hayan quedado repetidos (S_i y S_{i+1} idénticos). El hecho de que la nueva familia $(S'_i)_{i=1}^{2(|V'|-N(u))+1}$ cumple las tres propiedades de la descomposición de caminos se sigue directamente de que $(S_i)_{i=1}^{2|V'|+1}$ es una descomposición de caminos de $G[V']$. Por otro lado, es fácil comprobar que se mantiene la normalización. Si $w \in N(u)$ pertenece a S_i y S_{i-1} consecutivos, no se altera la normalización. Si $w \in N(u)$ pertenece a S_i y S_j no consecutivos, por la propiedad (2.7) debe cumplirse $u \in S_k$, $k = i, \dots, j$, por lo que tampoco se altera la normalización en este caso. Si $w \in N(u)$ pertenece a S_i para un solo índice i entonces, como la descomposición está normalizada, $S_{i-1} = S_i \setminus \{w\}$ y $S_{i+1} = S_i \setminus \{w\}$. Al eliminar w de la descomposición nos quedarían tres conjuntos iguales $S_{i-1} = S_i = S_{i+1} = S_i \setminus \{w\}$. Para conservar la normalización, será suficiente entonces eliminar los conjuntos consecutivos que hayan quedado repetidos como resultado de la eliminación de los nodos de $N(u)$. La descomposición $(S'_i)_{i=1}^{2(|V'|-N(u))+1}$ se emplea para el cálculo del empaquetamiento de mayor peso en $G[V' \setminus N(u)]$, proceso que se describe en la última parte de esta subsección.

Supongamos que este último proceso se ha completado, de manera que el coeficiente π_u ha sido obtenido. Si este coeficiente es positivo, es necesario actualizar la descomposición $(S_i)_{i=1}^{2|V'|+1}$. Para ello, se identifican los índices j, k tales que

$$j = \text{mín}\{i | (u, v) \in E, v \in S_i\}, \quad k = \text{máx}\{i | (u, v) \in E, v \in S_i\}.$$

A continuación, se añade u a todos los conjuntos S_i para $i \in \{j, \dots, k\}$. Ver que la familia resultante sigue siendo una descomposición de caminos se reduce a una mera comprobación de las condiciones (2.5), (2.6) y (2.7). Para completar la actualización, es necesario normalizar la descomposición resultante para que conjuntos consecutivos difieran en un nodo. Para ello, se intercala entre los conjuntos de índices $j - 1$ y j (respectivamente k y $k + 1$) el conjunto S_j

(respectivamente S_k) de la descomposición original.

Ejemplo 2.3.3. Supongamos que se desea levantar la desigualdad $\sum_{i=0}^4 x_i \leq 2$ correspondiente al ciclo de cinco nodos $H = (v_0, v_1, v_2, v_3, v_4, v_0)$. Sea $u \in V$ el nodo del levantamiento, cuyo conjunto de vecinos es $N(u) = \{v_1, v_2, v_3\}$. Entonces, una descomposición de caminos para $G[\{v_0, v_4\}]$ se obtiene eliminando v_1, v_2, v_3 de los conjuntos S_i obtenidos en el Ejemplo 2.3.2:

$$\begin{aligned} S_1 &= \emptyset, & S_5 &= \{v_0\}, & S_9 &= \{v_0, v_4\}, \\ S_2 &= \{v_0\}, & S_6 &= \{v_0\}, & S_{10} &= \{v_0\}, \\ S_3 &= \{v_0\}, & S_7 &= \{v_0\}, & S_{11} &= \emptyset, \\ S_4 &= \{v_0\}, & S_8 &= \{v_0, v_4\}, \end{aligned}$$

y eliminando los conjuntos consecutivos que se repitan:

$$\begin{aligned} S_1 &= \emptyset, & S_4 &= \{v_0\}, \\ S_2 &= \{v_0\}, & S_5 &= \emptyset. \\ S_3 &= \{v_0, v_4\}, \end{aligned}$$

Un empaquetamiento de peso máximo en $G[\{v_0, v_4\}]$ es $\{v_0\}$, por lo que el coeficiente correspondiente al levantamiento será $\pi_u = 2 - 1 = 1 > 0$. Para actualizar la descomposición del Ejemplo 2.3.2 de manera que sea una descomposición en el grafo $G[H \cup \{u\}]$, calculamos los índices j, k :

$$j = \min\{i \mid (u, v) \in E, v \in S_i\} = \min\{3, 4, 5, 6, 7, 8\} = 3$$

$$k = \max\{i \mid (u, v) \in E, v \in S_i\} = \max\{3, 4, 5, 6, 7, 8\} = 8$$

y añadimos u a los conjuntos que corresponden:

$$\begin{aligned} S_1 &= \emptyset, & S_5 &= \{v_0, v_2, u\}, & S_9 &= \{v_0, v_4\}, \\ S_2 &= \{v_0\}, & S_6 &= \{v_0, v_2, v_3, u\}, & S_{10} &= \{v_0\}, \\ S_3 &= \{v_0, v_1, u\}, & S_7 &= \{v_0, v_3, u\}, & S_{11} &= \emptyset. \\ S_4 &= \{v_0, v_1, v_2, u\}, & S_8 &= \{v_0, v_3, v_4, v\}, \end{aligned}$$

Finalmente se normaliza la descomposición obtenida:

$$\begin{aligned}
S_1 &= \emptyset, & S_6 &= \{v_0, v_2, u\}, & S_{11} &= \{v_0, v_4\}, \\
S_2 &= \{v_0\}, & S_7 &= \{v_0, v_2, v_3, u\}, & S_{12} &= \{v_0\}, \\
S_3 &= \{v_0, v_1\}, & S_8 &= \{v_0, v_3, u\}, & S_{13} &= \emptyset. \\
S_4 &= \{v_0, v_1, u\}, & S_9 &= \{v_0, v_3, v_4, u\}, & & \\
S_5 &= \{v_0, v_1, v_2, u\}, & S_{10} &= \{v_0, v_3, v_4\}, & &
\end{aligned}$$

En general, el proceso de levantamiento de la faceta asociada al agujero H descrito puede resumirse mediante el siguiente esquema:

0. Sean $V' = H$, $\sum_{v \in V'} \pi_v x_v \leq \pi_0$ faceta de $G[V']$ con $\boldsymbol{\pi}^{|V'|} = \mathbf{1}$ y $(S_i)_{i=1}^{2^{|V'|+1}}$ descomposición de caminos en $G[V']$ obtenida por la fórmula (2.8).
1. Sea $u \in V \setminus V'$. Obtener una descomposición de caminos de $G[V' \setminus N(u)]$ a partir de $(S_i)_{i=1}^{2^{|V'|+1}}$.
2. Utilizar la descomposición anterior para obtener w^u , el peso del empaquetamiento de menor peso en $G[V' \setminus N(u)]$.
3. Añadir a $\boldsymbol{\pi}$ la componente $\pi_u = \pi_0 - w^u$.
4. Actualizar $(S_i)_{i=1}^{2^{|V'|+1}}$, para que sea una descomposición en $G[V' \cup \{u\}]$.
5. $V' = V' \cup \{u\}$. Volver al paso 1. si $V' \neq V$.

A continuación se describe con detalle el desarrollo del paso 2., que por ser algo más complejo que el resto se ha dejado para el final.

Empaquetamientos de peso máximo por descomposición de caminos

Supongamos que tenemos un grafo $G = (V, E)$, una descomposición de caminos normalizada $(S_i)_{i=1}^{2^{|V|+1}}$ de G y un vector de pesos $\boldsymbol{\pi} \in \mathbb{R}^V$. Se desea obtener el empaquetamiento de mayor peso de G , es decir, que maximice la suma $\sum_{i \in V} \pi_v x_v$, donde $x_v = 1$ si v pertenece al empaquetamiento. Denotaremos con \mathcal{I} la colección de todos los empaquetamientos de G , mientras que $\mathcal{I}_i = \{I \in \mathcal{I} \mid I \subseteq S_i\}$ serán todos los empaquetamientos en $G[S_i]$. Finalmente, sea $V_i = \bigcup_{j=1}^i S_j$. Como la descomposición $(S_i)_{i=1}^{2^{|V|+1}}$ está normalizada, para cada $i > 1$ existe un nodo $v \in V$ tal que o bien $S_i = S_{i-1} \cup \{v\}$ o bien $S_i = S_{i-1} \setminus \{v\}$. Como consecuencia, los conjuntos \mathcal{I}_i satisfacen las siguiente relación recursiva:

$$\mathcal{I}_i = \begin{cases} \{\emptyset\} & \text{si } i = 1, \\ \mathcal{I}_{i-1} \cup \{(I \cup \{v\}) \in \mathcal{I} \mid I \in \mathcal{I}_{i-1}\} & \text{si } i > 1 \text{ y } S_i = S_{i-1} \cup \{v\}, \\ \{I \in \mathcal{I}_{i-1} \mid v \notin I\} & \text{si } i > 1 \text{ y } S_i = S_{i-1} \setminus \{v\}. \end{cases}$$

Dada la descomposición de caminos normalizada $(S_i)_{i=1}^{2^{|V|+1}}$ y el vector de pesos π , Verweij, 2000 calcula el peso del empaquetamiento de máximo peso con respecto a π mediante un algoritmo de programación dinámica. La programación dinámica se suele utilizar para resolver problemas de optimización mediante la combinación de soluciones para subproblemas más pequeños. Esta técnica no utiliza recursividad, sino que los resultados de los subproblemas se van almacenando y combinando para resolver problemas mayores, llegando hasta el tamaño deseado mediante un proceso iterativo. La programación dinámica se basa en el principio de optimalidad de Bellman: cualquier subsecuencia de una secuencia óptima debe ser, a su vez, una secuencia óptima. En el caso que nos ocupa, la programación dinámica no se aplica directamente sobre el problema del empaquetamiento de mayor peso, sino que se aplica sobre el problema de optimización que viene expresado por la siguiente función:

$$z_i : \mathcal{I}_i \rightarrow \mathbb{N} : I \mapsto \max \left\{ \sum_{v \in I'} \pi_v \mid I' \text{ es un empaquetamiento en } G[V_i], I' \cap S_i = I \right\}.$$

De la condición (2.5) se sigue que el valor $z_{2^{|V|+1}}(\emptyset)$ es el máximo peso de un empaquetamiento de G , es decir, el valor que se desea calcular. Por la definición de z_i tenemos que $z_1(\emptyset) = 0$. Para construir el algoritmo de programación dinámica es necesario encontrar una forma de combinar los valores de la función z_{i-1} para hallar los de z_i . Así, partiendo de $z_1(\emptyset) = 0$, de forma iterativa se hallará el valor $z_{2^{|V|+1}}(\emptyset)$ buscado. Supongamos entonces que $i > 1$. Sea $I \in \mathcal{I}_i$ un empaquetamiento en $G[S_i]$ y sea I^* el empaquetamiento de mayor peso en $G[V_i]$, con $I^* \cap S_i = I$. A continuación se analiza la relación de I^* con un empaquetamiento de máximo peso I' en $G[V_{i-1}]$, y se caracteriza $I' \cap S_{i-1}$. Consideramos cuatro casos:

- **Caso (i):** $S_i = S_{i-1} \cup \{v\}$ y $v \notin I$. Como $I^* \cap S_i = I$, debe ser $v \notin I^*$. Por lo tanto, I^* es también un empaquetamiento de peso máximo en $G[V_{i-1}]$ con $I^* \cap S_{i-1} = I$. Tomamos entonces $I' = I^*$ y $I' \cap S_{i-1} = I$.
- **Caso (ii):** $S_i = S_{i-1} \cup \{v\}$ y $v \in I$ (en particular $v \in I^*$). Afirmamos que

$$u \in S_i \quad \text{para toda arista } (u, v) \in E, \quad u \in V_i. \quad (2.9)$$

De esta afirmación se sigue que $I' = I^* \setminus \{v\}$ es un empaquetamiento de peso máximo con la condición de que $I' \cap S_{i-1} = I \setminus \{v\}$. Supongamos, por reducción al absurdo, que I'' es un empaquetamiento en $G[V_{i-1}]$ de mayor peso que I' y que además $I'' \cap S_{i-1} = I \setminus \{v\}$. Como $v \in I = I^* \cap S_{i-1}$, también $v \in I^*$, por lo que para cada $u \in V_i$ tal que $(u, v) \in E$ se tiene que $u \notin I^*$. Esto implica que cualquier u de esta forma no pertenece a I , pero según la afirmación (2.9) $u \in S_{i-1}$ y además $I'' \cap S_{i-1} = I \setminus \{v\}$. Se sigue entonces que $u \notin I''$ para toda arista $(u, v) \in E$, $u \in V_i$. Se puede añadir entonces v a I'' y encontrar un empaquetamiento en $G[V_i]$ de mayor peso que I^* .

Para probar la afirmación (2.9) procedamos por reducción al absurdo, suponiendo que existe $u \in V_i$ tal que $(u, v) \in E$ pero $u \notin S_i$. Se sigue que $u \in S_k$ para algún $k < i$. Por la condición (2.6) existe $j \in \{1, \dots, 2|V| + 1\}$ tal que $(u, v) \subseteq S_j$. Como $S_i = S_{i-1} \cup \{v\}$, por la condición (2.7) debe ser $j > i$ (en otro caso $v \in S_{i-1}$). Por otro lado, la condición (2.7) junto con $u \in S_k$ y $u \in S_j$ implica que $u \in S_i$, lo cual es una contradicción.

- **Caso (iii):** $S_i = S_{i-1} \setminus \{v\}$ y $I \cup \{v\}$ es un empaquetamiento en $G[S_{i-1}]$. En este caso $V_i = V_{i-1}$, por lo que $I' = I^*$ es un empaquetamiento de peso máximo en $G[V_{i-1}]$ con $I' \cap S_i = I$. O bien $v \in I'$, en cuyo caso $I' \cap S_{i-1} = I \cup \{v\}$, o bien $v \notin I'$ y $I' \cap S_{i-1} = I$.
- **Caso (iv):** $S_i = S_{i-1} \setminus \{v\}$ y $I \cup \{v\}$ no es un empaquetamiento en $G[S_{i-1}]$. Observamos que la igualdad $I^* \cap S_i = I$ implica que $(I^* \cup \{v\}) \cap S_{i-1} = I \cup \{v\}$. Como I^* es un empaquetamiento en $G[V_i]$, $I^* \cap S_{i-1}$ lo es en $G[S_{i-1}]$. Si $v \in I^*$ tendríamos $I^* \cap S_{i-1} = (I^* \cup \{v\}) \cap S_{i-1} = I \cup \{v\}$, y llegaríamos a una contradicción al no ser $I \cup \{v\}$ un empaquetamiento. Por tanto, $v \notin I^*$. Podemos tomar entonces $I' = I^*$ como en el caso anterior y además sabemos que como $v \notin I'$, $I' \cap S_{i-1} = I$.

El análisis de casos anterior lleva a la siguiente expresión recurrente para $z_i(I)$, con $i > 1$:

$$z_i(I) = \begin{cases} z_{i-1}(I) & \text{si } S_i = S_{i-1} \cup \{v\} \text{ y } v \notin I, \\ z_{i-1}(I \setminus \{v\}) + \pi_v & \text{si } S_i = S_{i-1} \cup \{v\} \text{ y } v \in I, \\ \max\{z_{i-1}(I), z_{i-1}(I \cup \{v\})\} & \text{si } S_i = S_{i-1} \setminus \{v\} \text{ y } I \cup \{v\} \in \mathcal{I}_{i-1}, \\ z_{i-1}(I) & \text{si } S_i = S_{i-1} \setminus \{v\} \text{ y } I \cup \{v\} \notin \mathcal{I}_{i-1}. \end{cases} \quad (2.10)$$

Ejemplo 2.3.4. Sea $G = (V, E)$ el grafo de la Figura 2.7. Partimos de un agujero impar $H = (v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_0)$ y su faceta asociada $\sum_{i=0}^6 x_i \leq 3$. Sea $v \in V$, $N(v) =$

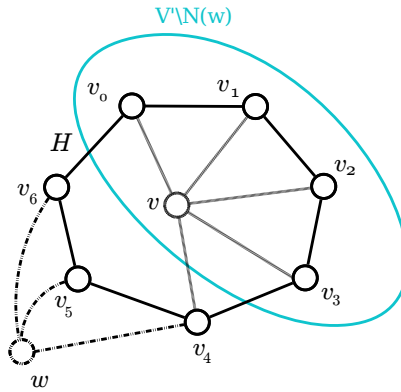


Figura 2.7: Levantamiento de un agujero impar

$\{v_0, v_1, v_2, v_3, v_4\}$. La faceta levantada al introducir el nuevo nodo v es

$$2x_v + \sum_{i=0}^6 x_i \leq 3, \quad (2.11)$$

pues un empaquetamiento de peso máximo en $H \setminus N(v)$ es $\{v_5\}$ o $\{v_6\}$. El conjunto de nodos “explorados” es $V' = \{v_0, v_1, v_2, v_3, v_4, v_5, v_6, v\}$. Supongamos ahora que queremos levantar la faceta (2.11), añadiendo un nodo w con $N(w) = \{v_4, v_5, v_6\}$ (ver Figura 2.7). Para ello, aplicamos el algoritmo de programación dinámica descrito anteriormente. Partimos de una descomposición de caminos del subgrafo $G[V' \setminus N(w)]$:

$$\begin{aligned} S_1 &= \emptyset, & S_5 &= \{v_0, v_1, v_2, v\}, & S_9 &= \{v_0, v_3\}, \\ S_2 &= \{v_0\}, & S_6 &= \{v_0, v_2, v\}, & S_{10} &= \{v_0\}, \\ S_3 &= \{v_0, v\}, & S_7 &= \{v_0, v_2, v_3, v\}, & S_{11} &= \emptyset. \\ S_4 &= \{v_0, v_1, v\}, & S_8 &= \{v_0, v_3, v\}, \end{aligned}$$

La Tabla 2.1 muestra el desarrollo del algoritmo de programación dinámica. Cada fila se corresponde con un índice $i = 1, \dots, 11$, mientras que las columnas se refieren a conjuntos independientes del grafo $G[V' \setminus N(w)]$. Así, cada entrada (i, I) de esta tabla muestra el valor $z_i(I)$. El dominio de las funciones z_i es distinto en cada índice, quedando vacías las casillas (i, I) para las que no exista el valor $z_i(I)$. El valor buscado es $z_{11}(\emptyset) = 2$, de donde el coeficiente del levantamiento es $\pi_w = 3 - 2 = 1$. Para construir la Tabla 2.1, partiendo de $z_1(\emptyset) = 0$, se emplea la fórmula (2.10). A continuación ilustramos esta construcción, tomando como ejemplo algunas de las casillas de la tabla:

	\emptyset	$\{v_0\}$	$\{v_1\}$	$\{v_2\}$	$\{v_3\}$	$\{v\}$	$\{v_0, v_2\}$	$\{v_0, v_3\}$
1	0							
2	0	0+1						
3	0	1				0+2		
4	0	1	0+1			2		
5	0	1	1	0+1		2	1+1	
6	$\max\{0, 1\}$	1		1		2	2	
7	1	1		1	1+1	2	2	1+1
8	$\max\{1, 1\}$	$\max\{1, 2\}$			2	2		2
9	$\max\{1, 2\}$	2			2			2
10	$\max\{2, 2\}$	$\max\{2, 2\}$						
11	$\max\{2, 2\}$							

Table 2.1: Valores de la función z_i , $i = 1, \dots, 11$

- **$i = 2$:** En este caso $S_2 = S_1 \cup \{v_0\}$, por lo que habrá que aplicar alguno de los dos primeros supuestos de la fórmula (2.10).
 - $z_2(\emptyset) : v_0 \notin \emptyset$ por lo que $z_2(\emptyset) = z_1(\emptyset) = 0$.
 - $z_2(\{v_0\}) : v_0 \in \{v_0\}$ por lo que $z_2(\{v_0\}) = z_1(\{v_0\} \setminus \{v_0\}) + \pi_0 = z_1(\emptyset) + \pi_0 = 0 + 1 = 1$.
- **$i = 3$:** Tenemos $S_3 = S_2 \cup \{v\}$.
 - $z_3(\emptyset) : v \notin \emptyset$ por lo que $z_3(\emptyset) = z_2(\emptyset) = 0$.
 - $z_3(\{v_0\}) : v \notin \{v_0\}$ por lo que $z_3(\{v_0\}) = z_2(\{v_0\}) = 1$.
 - $z_3(\{v\}) : \text{como } v \in \{v\}, z_3(\{v\}) = z_2(\{v\} \setminus \{v\}) + \pi_v = z_2(\emptyset) + \pi_v = 0 + 2 = 2$.
- **$i = 6$:** En este caso $S_6 = S_5 \setminus \{v_1\}$. Tendremos que aplicar alguno de los dos últimos supuestos de la fórmula (2.10).
 - $z_6(\emptyset) : \emptyset \cup \{v_1\}$ es conjunto independiente de S_5 por lo que $z_6(\emptyset) = \max\{z_5(\emptyset), z_5(\{v_1\})\} = \max\{0, 1\} = 1$.
 - $z_6(\{v_0\}) : \text{Como } \{v_0\} \cup \{v_1\}$ no es conjunto independiente de S_5 , $z_6(\{v_0\}) = z_5(\{v_0\}) = 1$.

Ejemplo 2.3.5. Partiendo del grafo de la Figura 2.7 y de la descomposición de caminos para $G[V' \setminus N(w)]$ dada en el ejemplo anterior, vamos a ilustrar el análisis de casos que se realizó para obtener la expresión (2.10). Para cada caso (i) – (iv), identificamos los conjuntos I^* e I' :

- **Caso (i):** consideramos el conjunto $S_7 = S_6 \cup \{v_3\}$ y el conjunto independiente de S_7 $I = \{v_0, v_2\}$. El conjunto independiente I^* de máximo peso en $G[V_7] = G[V' \setminus N(w)]$ tal que $I^* \cap S_7 = I$ es $I^* = I$. En este caso $I' = I^*$ es también conjunto independiente de peso máximo en $G[V_6] = G[V_7 \setminus \{v_3\}]$ y $I' \cap S_6 = I$.
- **Caso (ii):** consideramos de nuevo $S_7 = S_6 \cup \{v_3\}$. Tomamos ahora $I = \{v_0, v_3\}$ conjunto independiente de S_7 . El conjunto independiente I^* de máximo peso en $G[V_7] = G[V' \setminus N(w)]$ tal que $I^* \cap S_7 = I$ es $I^* = I$. El conjunto independiente I' de peso máximo en $G[V_6] = G[V_7 \setminus \{v_3\}]$ cumpliendo $I' \cap S_6 = I \setminus v_3$ es $I' = \{v_0\}$.
- **Caso (iii):** tomamos $S_8 = S_7 \setminus \{v_2\}$ e $I = \{v_0\}$ un conjunto independiente en S_8 . El conjunto independiente I^* de máximo peso en $G[V_8] = G[V_7]$ tal que $I^* \cap S_8 = I$ es $I^* = \{v_0, v_2\}$. Podemos tomar $I' = I^*$, $I' \cap S_{i-1} = I \cup \{v_2\}$.
- **Caso (iv):** tomamos $S_8 = S_7 \setminus \{v_2\}$ e $I = \{v_0, v_3\}$ un conjunto independiente en S_8 . El conjunto independiente I^* de máximo peso en $G[V_8] = G[V_7]$ tal que $I^* \cap S_8 = I$ es $I^* = I$. Podemos tomar $I' = I^*$, $I' \cap S_{i-1} = I$.

Capítulo 3

Etiquetado por redes de restricción

Este capítulo presenta un resumen de los aspectos más destacados del quinto capítulo de la tesis de Klau (Klau, 2001), dedicado al etiquetado de mapas. En su tesis, Klau considera el caso en que las etiquetas son rectángulos orientados con los ejes de coordenadas, es decir, horizontal o verticalmente. Partiendo de esta base, se modela el problema del etiquetado a través de una clase especial de grafos dirigidos, las llamadas redes de restricción. Una solución a un problema de etiquetado vendrá codificada a través de asignaciones de valores a los nodos del grafo. En el presente capítulo, una vez descrito este original modelado del problema, veremos cómo se utiliza para obtener el equivalente a los distintos tipos de problema de etiquetado que fueron presentados en la introducción (puro, maximización de etiquetas, minimización de solapamiento, etc.). Finalmente, basadas en las ideas anteriores, se presentarán algunas formulaciones de optimización lineal entera propuestas por Klau.

El trabajo desarrollado por Klau abarca todos los supuestos de distribución de las etiquetas presentados en la Figura 3.1. Entre los modelos representados se distinguen dos tipos: combinatorios, con un número finito de posiciones para las etiquetas, y continuos, con deslizamiento de las etiquetas. Como muestra la figura, en todos los casos las etiquetas contienen en alguna de sus fronteras al punto que describen. Se consideran tres modelos combinatorios en los que la posición de una etiqueta determinada está fija, a elegir entre 4, 2 o 1 opciones. En los modelos con deslizamiento se permite el movimiento de las etiquetas sobre el punto correspondiente en las direcciones indicadas en la Figura 3.1. En esta figura, el área sobre la que se puede colocar la etiqueta se delimita con un trazado discontinuo. En el modelo “4-deslizamiento” se consideran cuatro etiquetas que pueden deslizarse en dirección horizontal o vertical siempre que el punto asociado no salga de la frontera de la etiqueta. El modelo “2-deslizamiento” es similar, con la

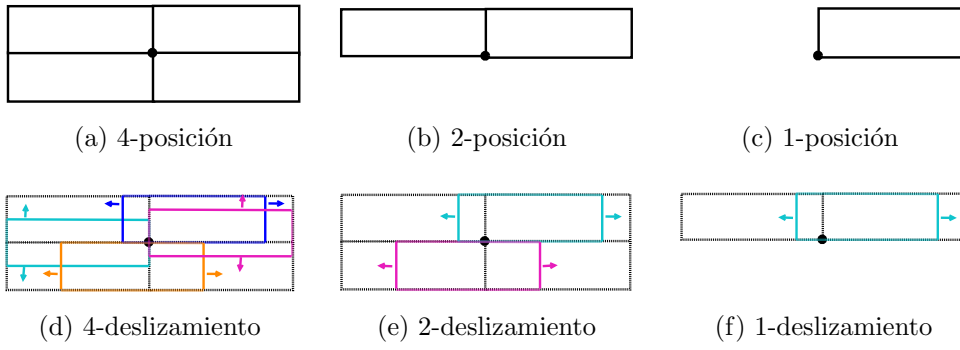


Figura 3.1: Modelos según las posibilidades para colocar las etiquetas

diferencia de que sólo se permiten deslizamientos horizontales de las etiquetas. Finalmente, el modelo “1-deslizamiento” considera dos etiquetas posibles, con desplazamiento horizontal.

3.1. Conceptos previos

En esta sección se introducen los conceptos de red de restricción y asignación factible, que van a ser fundamentales en el desarrollo del capítulo. Además, el resultado presentado por el Teorema 3.1.1, es la base teórica que sustenta el modelado llevado a cabo por Klau. Para poder entender estos nuevos conceptos es conveniente recordar algunas nociones básicas sobre grafos y redes dirigidas.

Definición 3.1.1. *Un grafo dirigido $G = (V, A)$ es una estructura formada por un conjunto finito de nodos, V , y un conjunto de arcos o pares ordenados de nodos, $A \subseteq \{a = (u, v) | u, v \in V\}$. Dado $a = (u, v) \in A$ decimos que u es el nodo inicial del arco y v el nodo final.*

Definición 3.1.2. *Un camino en un grafo dirigido $G = (V, A)$ es una sucesión de nodos (v_1, v_2, \dots, v_k) de manera que $(v_i, v_{i+1}) \in A$ para cada $i = 1, \dots, k - 1$. Si $v_1 = v_k$ entonces el camino se denomina circuito.*

Definición 3.1.3. *Una red dirigida $R = (V, A, w)$ es un grafo dirigido $G = (V, A)$ con una función de peso que asocia a cada arco del grafo un valor, $w : A \rightarrow \mathbb{R}$.*

Definición 3.1.4. *Una asignación de valores a los nodos de una red $R = (V, A, w)$, $c : V \rightarrow \mathbb{R}$, se dice factible si satisface para cada arco $a = (u, v) \in A$ la desigualdad*

$$c(v) - c(u) \geq w(a).$$

En este contexto, cuando la función de pesos de la red se utiliza para codificar este tipo de desigualdades, hablamos de red de restricción.

Observación 1. Sea $R = (V, A, w)$ una red de restricción y c una asignación factible para R . Sea $P = (v_1, v_2, \dots, v_k)$ un camino en A . Se verifica la siguiente desigualdad:

$$c(v_k) - c(v_1) \geq \sum_{i=1}^{k-1} w((v_i, v_{i+1})).$$

Teorema 3.1.1 (Cook et al., 1998). *Dada una red de restricción $R = (V, A, w)$, existe una asignación factible asociada a ella si y solo si A no contiene un circuito con peso total positivo.*

3.2. Redes de etiquetado

En esta sección se detalla el modelado de los componentes del problema del etiquetado así como sus restricciones, a través de las redes de etiquetado.

Definición 3.2.1. *Un par de redes de etiquetado son dos redes de restricción $R_x = (V_x, A_x, w_x)$ y $R_y = (V_y, A_y, w_y)$ que corresponden a una instancia $(\mathcal{P} = \{p_1, \dots, p_n\}, \{L_i\}_{i \in I})$ del problema de etiquetado. Los conjuntos de nodos de R_x y R_y son*

$$V_x = \{x_i | i \in I\} \cup \{l_i | i \in I\} \cup \{r_i | i \in I\}$$

$$V_y = \{y_i | i \in I\} \cup \{b_i | i \in I\} \cup \{t_i | i \in I\}.$$

Los nodos x_i e y_i representan las coordenadas en el plano de los puntos a etiquetar. Los nodos l_i, r_i se corresponden con las coordenadas horizontales de las fronteras izquierda y derecha de la etiqueta asignada al punto p_i , respectivamente. Finalmente, b_i y t_i representan respectivamente las coordenadas verticales de las fronteras inferior y superior de esta etiqueta. De esta manera, un par (c_x, c_y) de asignaciones para las redes R_x y R_y determinan completamente una posible distribución de las etiquetas, y por tanto una solución al problema del etiquetado. Para que esta solución cumpla las restricciones requeridas, se eligen los arcos A_x, A_y y los pesos w_x, w_y convenientemente. Nos referiremos a un par (c_x, c_y) de asignaciones factibles para las redes R_x y R_y con el término *asignación de coordenadas*. La Figura 3.2 ilustra la correspondencia entre los nodos de la red y los elementos del problema de etiquetado.

A continuación, se describen conjuntos de arcos que modelan distintos aspectos del problema de etiquetado. En primer lugar, se fija la distribución relativa de los puntos a etiquetar en

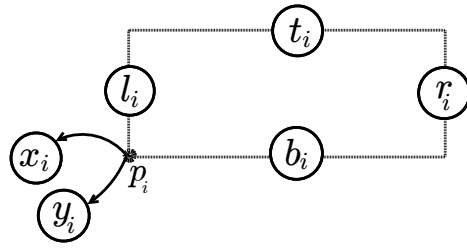


Figura 3.2: Nodos de la red de restricción asociados a un punto p_i .

el plano, con los *arcos de distancia fija*. Después, se modelan las etiquetas: los *arcos de tamaño de etiqueta* describen el rectángulo correspondiente a una etiqueta y los *arcos de proximidad* aseguran que un punto será cercano a la etiqueta que lo describe. Los *arcos frontera* son inversos a los arcos de proximidad y evitan que un punto quede en el interior de la etiqueta que lo describe. Finalmente, los *arcos de separación de etiquetas* controlan los solapamientos entre etiquetas.

Modelado de los puntos

Sean (α_i, β_i) , $i \in I$, las coordenadas de cada punto que se desea etiquetar. La posición de los puntos se fija insertando en las redes R_x y R_y arcos formando los siguientes caminos

$$P_x = (x_1, \dots, x_n) \quad P_{-x} = (x_n, \dots, x_1)$$

$$P_y = (y_1, \dots, y_n) \quad P_{-y} = (y_n, \dots, y_1)$$

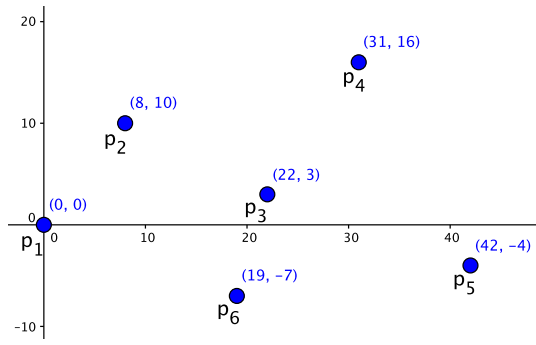
con pesos

$$w_x((x_i, x_{i+1})) = \alpha_{i+1} - \alpha_i \quad w_x((x_{i+1}, x_i)) = \alpha_i - \alpha_{i+1}$$

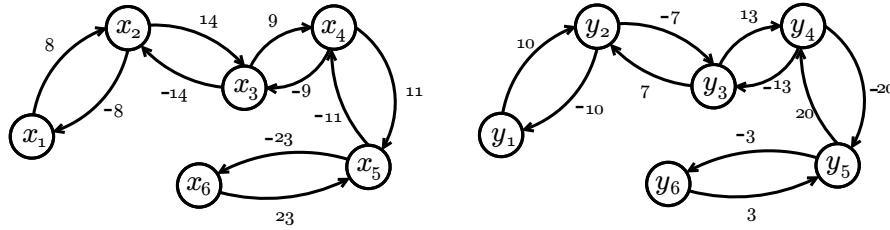
$$w_y((y_i, y_{i+1})) = \beta_{i+1} - \beta_i \quad w_y((y_{i+1}, y_i)) = \beta_i - \beta_{i+1}$$

para cada $i \in \{1, \dots, n-1\}$. Los arcos de estos caminos se denominan *arcos de distancia fija* y nos referiremos al conjunto de todos ellos como A_F . La Figura 3.3 ilustra la construcción de las redes de etiquetado con estos arcos.

Lema 3.2.1. Sean $R_x = (V_x, A_x, w_x)$ y $R_y = (V_y, A_y, w_y)$ un par de redes de etiquetado para una instancia $(\mathcal{P}, \{L_i\}_{i \in I})$ del problema de etiquetado. Sea $A_F \subseteq A_x \cup A_y$ el conjunto de arcos de distancias fijas. Una asignación de coordenadas c que respete las restricciones codificadas en A_F resulta en una correcta distribución de los puntos en el plano (salvo traslación).



(a) Disposición de los puntos



(b) Arcos de distancia fija

Figura 3.3: Ejemplo de construcción de los arcos de distancia fija

Demostración. Sean $P_x = (x_1, \dots, x_n)$, $P_{-x} = (x_n, \dots, x_1)$, $P_y = (y_1, \dots, y_n)$ y $P_{-y} = (y_n, \dots, y_1)$ los caminos asociados a los arcos de A_F . Dado que la asignación c es factible, para todo $i \in \{1, \dots, n-1\}$ se cumple que $c(x_{i+1}) - c(x_i) \geq \alpha_{i+1} - \alpha_i$. Por la misma razón también debe verificarse que $c(x_i) - c(x_{i+1}) \geq \alpha_i - \alpha_{i+1}$ para todo $i \in \{1, \dots, n-1\}$. Se concluye que:

$$c(x_{i+1}) - c(x_i) = \alpha_{i+1} - \alpha_i, \quad \forall i \in \{1, \dots, n-1\}. \quad (3.1)$$

Lo mismo ocurre para las coordenadas y :

$$c(y_{i+1}) - c(y_i) = \beta_{i+1} - \beta_i, \quad \forall i \in \{1, \dots, n-1\}. \quad (3.2)$$

Como cada nodo en V_x (respect. V_y) está unido con cualquier otro por dos subcaminos inversos de P_x y P_{-x} (respect. P_y y P_{-y}), la correcta posición relativa de los nodos se sigue de la Observación 1. Es, decir, las igualdades (3.1) y (3.2) se cumple también para cualquier par de nodos no consecutivos en los caminos P_x y P_y respectivamente. Por tanto, como (α_i, β_i) son las coordenadas del punto p_i , tenemos que la distribución de los puntos según la asignación c coincide salvo traslación con la distribución original. \square

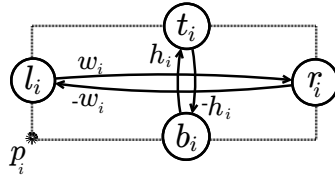


Figura 3.4: Arcos de tamaño de la etiqueta asociada a un punto p_i .

Modelado de las etiquetas

Para cada $i \in I$, una etiqueta $j \in L_i$ representa un rectángulo de anchura w_i y altura h_i , situado en una posición determinada alrededor de p_i . Para modelar el tamaño de este rectángulo se introducen cuatro *arcos de tamaño de etiqueta* para cada $i \in I$,

$$A_L(i) = \{(l_i, r_i), (r_i, l_i), (b_i, t_i), (t_i, b_i)\}.$$

Los pesos de estos arcos son

$$\begin{aligned} w_x((l_i, r_i)) &= w_i, & w_x((r_i, l_i)) &= -w_i, \\ w_y((b_i, t_i)) &= h_i, & w_y((t_i, b_i)) &= -h_i. \end{aligned}$$

Además de codificar el tamaño de los rectángulos, es necesario asegurar que la etiqueta se sitúa cercana al punto que describe. Sea $i \in I$, se definen los *arcos de proximidad* (ver Figura 3.5) como

$$A_P(i) = \{(x_i, r_i), (l_i, x_i), (y_i, t_i), (b_i, y_i)\}.$$

Estos arcos tienen peso cero, de manera que se excluyen colocaciones de la etiqueta que dejan a p_i fuera del rectángulo de su etiqueta. Para ver esto, basta con escribir las restricciones correspondientes a estos nuevos arcos para una asignación $c = (c_x, c_y)$:

$$\begin{aligned} c_x(r_i) - c_x(x_i) &\geq 0, & c_x(x_i) - c_x(l_i) &\geq 0, \\ c_y(t_i) - c_y(y_i) &\geq 0, & c_y(y_i) - c_y(b_i) &\geq 0. \end{aligned}$$

Reescribiendo las desigualdades anteriores tenemos que la asignación de coordenadas $(c_x(x_i), c_y(y_i))$

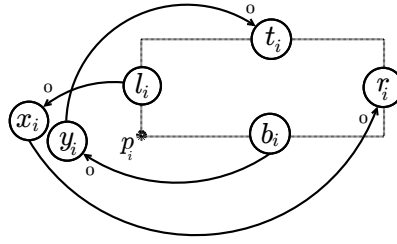


Figura 3.5: Arcos de proximidad para la etiqueta asociada a un punto p_i .

se encuentra dentro de los límites del rectángulo correspondiente, para cada $i \in I$:

$$c_x(l_i) \leq c_x(x_i) \leq c_x(r_i),$$

$$c_y(b_i) \leq c_y(y_i) \leq c_y(t_i).$$

Sin embargo, el punto $(c_x(x_i), c_y(y_i))$ todavía puede estar dentro del rectángulo, produciendo una situación que no es conforme al modelo del etiquetado. Dependiendo del tipo de modelo, se define una constante

$$d = \begin{cases} 1 & \text{en modelos con deslizamiento} \\ 2 & \text{en modelos discretos} \end{cases}$$

y se evita este tipo de situación, añadiendo al conjunto de arcos al menos d de los arcos en el siguiente conjunto de *arcos frontera*

$$A_B(i) = \begin{cases} \{(r_i, x_i), (x_i, l_i), (t_i, y_i), (y_i, b_i)\} & \text{en modelos 4-deslizamiento y 4-posición,} \\ \{(r_i, x_i), (x_i, l_i), (y_i, b_i)\} & \text{en el modelo 2-posición,} \\ \{(t_i, y_i), (y_i, b_i)\} & \text{en el modelo 2-deslizamiento,} \\ \{(x_i, l_i), (y_i, b_i)\} & \text{en el modelo 1-posición,} \\ \{(y_i, b_i)\} & \text{en el modelo 1-deslizamiento.} \end{cases}$$

Estos arcos tienen peso cero y son inversos de los arcos de proximidad. Cada arco de $\{(r_i, x_i), (x_i, l_i), (t_i, y_i), (y_i, b_i)\}$ se corresponde con una frontera del rectángulo asociado a la etiqueta de i . La adición de uno de estos arcos, junto con su arco de proximidad inverso, obligará a que el punto pertenezca a la frontera correspondiente. Si, por ejemplo, el arco (r_i, x_i) está presente en R_x esto obliga, junto con el arco de proximidad inverso (x_i, r_i) , a que la coordenada $c_x(r_i)$ sea igual a $c_x(x_i)$ para una asignación factible. Esto es, el punto p_i se situará sobre la frontera derecha de la etiqueta. Claramente, en un modelo con deslizamiento al menos uno de

los arcos $\{(r_i, x_i), (x_i, l_i), (t_i, y_i), (y_i, b_i)\}$ debe estar presente; puede ser cualquiera en el modelo 4-deslizamiento, debe ser un arco vertical $((y_i, b_i)$ o $(t_i, y_i))$ en el modelo 2-deslizamiento y debe ser (y_i, b_i) en el caso 1-deslizamiento. Para los modelos discretos, deben añadirse dos de estos arcos para asegurar que el punto p_i coincide con la esquina intersección de las dos fronteras correspondientes.

Lema 3.2.2. Sean $R_x = (V_x, A_x, w_x)$ y $R_y = (V_y, A_y, w_y)$ un par de redes de etiquetado para una instancia $(\mathcal{P}, \{L_i\}_{i \in I})$ del problema de etiquetado. Sea $I' \subseteq I$ un subconjunto de índices de puntos. Supongamos que A_x y A_y contienen los arcos de distancias fijas A_F y, para cada $i \in I'$, los arcos de tamaño de etiqueta $A_L(i)$, los de proximidad $A_P(i)$ y al menos d de los arcos frontera $B(i) \subseteq A_B(i)$, $|B(i)| = d$. Es decir:

$$A_x \cup A_y = A_F \bigcup_{i \in I'} (A_L(i) \cup A_P(i) \cup B(i)).$$

Entonces, una asignación de coordenadas c que respete $A_x \cup A_y$, resulta en un etiquetado de los puntos I' en el que cada punto p_i , $i \in I'$, se etiqueta con un rectángulo de anchura w_i y altura h_i . Cada punto yace en la frontera de su rectángulo si $d = 1$ y en una de sus esquinas si $d = 2$.

Demostración. A partir de la asignación de coordenadas c , construimos un etiquetado y vemos que se cumplen las propiedades del enunciado. Para cada $i \in I'$, c determina la coordenada inferior izquierda y superior derecha del rectángulo que lo etiqueta:

$$ii(i) = (c_x(l_i), c_y(b_i)),$$

$$sd(i) = (c_x(r_i), c_y(t_i)).$$

Como c respeta $A_L(i)$, tenemos que $sd(i)_x - ii(i)_x = w_i$ y $sd(i)_y - ii(i)_y = h_i$. Por otro lado, $B(i)$ contiene como mucho un arco horizontal y otro vertical, pues si contiene dos arcos del mismo tipo existiría un circuito positivo formado por estos dos y un arco de tamaño de etiqueta, lo cual contradiría la factibilidad de c por el Teorema 3.1.1. Como c respeta $A_P(i)$ tenemos que $sd(i)_x - c_x(x_i) \geq 0$, $c_x(x_i) - ii(i)_x \geq 0$, $sd(i)_y - c_y(y_i) \geq 0$ y $c_y(y_i) - ii(i)_y \geq 0$. Si $d = 1$, al menos una de estas desigualdades se convierte en igualdad y si $d = 2$ una desigualdad horizontal y otra vertical serán igualdades. Se sigue que p_i está en la frontera del rectángulo correspondiente si $d = 1$ y coincide con una esquina si $d = 2$. \square

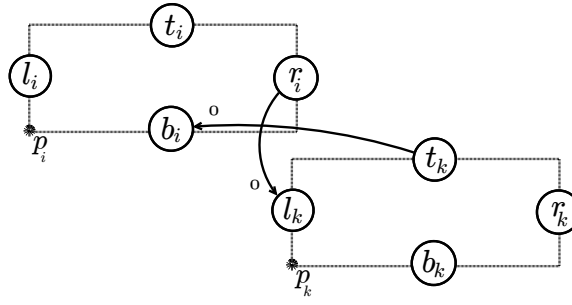


Figura 3.6: Arcos de separación entre las etiquetas de dos puntos p_i y p_k

Modelado de los solapamientos

Decimos que dos rectángulos disjuntos están verticalmente separados si uno está colocado por encima o por debajo del otro y que están horizontalmente separados si uno de ellos está situado a la izquierda o derecha del otro. Un par de rectángulos está separado si está horizontal o verticalmente separado. Dos rectángulos se solapan si no están separados. En el par de redes de etiquetado, evitamos los solapamientos con los *arcos de separación de etiquetas*. Para cada par de puntos $p_i, p_k, i, k \in I$, se definen los arcos de separación entre sus etiquetas como

$$A_S(i, k) \in \mathcal{P}(\{(t_k, b_i), (t_i, b_k), (r_k, l_i), (r_i, l_k)\}).$$

Estos arcos tienen peso cero.

La presencia de un arco de separación en las redes de etiquetado depende de la posición relativa de los pares de puntos. Si estos están suficientemente separados, sus etiquetas nunca se solaparán y no será necesario añadir sus correspondientes arcos de separación. Para cada par de puntos $p_i, p_k, i, k \in I$, se calculan las regiones donde pueden situarse sus respectivas etiquetas. Si la intersección de estas dos regiones es vacía, entonces $A_S(i, k) = \emptyset$. En caso contrario, dependiendo de la posición relativa de los puntos p_i y p_k , $A_S(i, k)$ contiene los siguientes arcos de separación:

1. Si $\alpha_k \geq \alpha_i$, $(r_i, l_k) \in A_S(i, k)$.
2. Si $\alpha_i \geq \alpha_k$, $(r_k, l_i) \in A_S(i, k)$.
3. Si $\beta_k \geq \beta_i$, $(t_i, b_k) \in A_S(i, k)$.
4. Si $\beta_i \geq \beta_k$, $(t_k, b_i) \in A_S(i, k)$.

La Figura 3.6 muestra los arcos de separación utilizados para un ejemplo concreto.

Lema 3.2.3. Sean $R_x = (V_x, A_x, w_x)$ y $R_y = (V_y, A_y, w_y)$ un par de redes de etiquetado para una instancia $(\mathcal{P}, \{L_i\}_{i \in I})$ del problema de etiquetado. Sea $I' \subseteq I$ un subconjunto de índices de puntos y sean $p_i, p_k, i, k \in I'$ dos puntos distintos. Supongamos A_x y A_y como en el Lema 3.2.2. Sea c una asignación de coordenadas que respeta $A_x \cup A_y$. Si el conjunto $A_S(i, k)$ es vacío o c respeta al menos uno de los arcos de $A_S(i, k)$, entonces las etiquetas de p_i y p_k no se solapan en el etiquetado.

Demostración. Por el Lema 3.2.2, sabemos que tenemos un etiquetado del conjunto de puntos I' . Distinguimos los dos casos siguientes:

- $A_S(i, k) = \emptyset$. Por construcción del conjunto de arcos de separación las etiquetas no se pueden solapar, pues los puntos están suficientemente separados.
- $A_S(i, k) \neq \emptyset$. En este caso la asignación de coordenadas respeta al menos un arco de separación. Si este arco es de tipo horizontal, la etiqueta de p_i estará a la izquierda o a la derecha de la de p_k , y estará arriba o abajo si el arco es vertical.

□

3.3. Definiciones alternativas de los problemas de etiquetado

Una vez presentada la construcción de las redes de etiquetado, Klau, 2001, reformula los problemas de etiquetado de las definiciones 1.2.3, 1.2.4 y 1.2.5 utilizando estas nuevas estructuras creadas. La idea consiste en partir de un par de redes de etiquetado que contienen los arcos de distancia fija, tamaño de etiqueta y proximidad, y encontrar una extensión de estas redes añadiendo arcos frontera y de separación. Nos referiremos a estos dos últimos con el nombre de arcos potenciales:

$$A_{pot} = \left(\bigcup_{i \in I} A_B(i) \right) \cup \left(\bigcup_{i, k \in I, i \neq k} A_S(i, k) \right).$$

En lo sucesivo, asumiremos que $d = 1$ para modelos con deslizamiento y $d = 2$ para modelos discretos. Sea $(\mathcal{P}, \{L_i\}_{i \in I})$ una instancia del problema del etiquetado, y sean $R_x = (V_x, A_x, w_x)$ y $R_y = (V_y, A_y, w_y)$ un par de redes de etiquetado que contienen los arcos de distancias fijas, de tamaño de etiqueta y de proximidad. En estas condiciones se tienen las siguiente definiciones alternativas de los problemas de etiquetado:

Definición 3.3.1. (Problema del etiquetado, ETI). Encontrar un subconjunto de arcos potenciales $A \subseteq A_{pot}$ con las propiedades:

- i) $|A \cap A_B(i)| \geq d$, para todo $i \in I$.
- ii) $|A \cap A_S(i, k)| \geq 1$, para todo $i, k \in I$, $i \neq k$, $A_S(i, k) \neq \emptyset$.
- iii) $A \cup A_x \cup A_y$ no contiene ningún circuito positivo.

Definición 3.3.2. (Problema de maximización del número de etiquetas, MNE). Encontrar un subconjunto de puntos $I' \subseteq I$ del mayor tamaño posible y subconjunto de arcos potenciales $A \subseteq A_{pot}$ con las propiedades:

- i) $|A \cap A_B(i)| \geq d$, para todo $i \in I'$.
- ii) $|A \cap A_S(i, k)| \geq 1$, para todo $i, k \in I'$, $i \neq k$, $A_S(i, k) \neq \emptyset$.
- iii) $A \cup A_x \cup A_y$ no contiene ningún circuito positivo.

Definición 3.3.3. (Problema de minimización del número de solapamientos, MNS). Encontrar un subconjunto de pares de puntos $I_{sep} \subseteq \binom{I}{2}$ del mayor tamaño posible y subconjunto de arcos potenciales $A \subseteq A_{pot}$ con las propiedades:

- i) $|A \cap A_B(i)| \geq d$, para todo $i \in I$.
- ii) $|A \cap A_S(i, k)| \geq 1$, para todo $(i, k) \in I_{sep}$.
- iii) $A \cup A_x \cup A_y$ no contiene ningún circuito positivo.

3.4. Formulaciones de optimización lineal entera

En esta sección presentamos varias formulaciones de optimización lineal entera propuestas por Klau para modelar algunas variantes del problema de etiquetado definidas en la sección anterior. En general, se desea describir una extensión del par de redes de etiquetado, añadiéndoles a estas un subconjunto de arcos de A_{pot} . Dichos arcos son variables de decisión en las distintas formulaciones y deben satisfacer unas determinadas propiedades que dependen de la variante con la que se trabaje.

Al igual que en la sección anterior, consideramos $R_x = (V_x, A_x, w_x)$ y $R_y = (V_y, A_y, w_y)$ un par de redes de etiquetado que contienen únicamente los arcos de distancias fijas, A_F , los de

tamaño de etiqueta, $\bigcup_{i \in I} A_S(i)$ y los de proximidad, $\bigcup_{i \in I} A_P(i)$. En esta sección, denotaremos con $\hat{A} \subseteq A_{pot}$ el conjunto de arcos potenciales que forman parte de la extensión de las redes R_x y R_y . Se definen las siguientes variables de decisión, $\mathbf{z} \in A_{pot}$:

$$z_a = \begin{cases} 1 & \text{si } a \in \hat{A}, \\ 0 & \text{en otro caso.} \end{cases}$$

3.4.1. Formulación para ETI

En las condiciones anteriores, sea $\mathbf{m} \in \mathbb{Q}^{A_{pot}}$ un vector de coeficientes arbitrarios. Se tiene la siguiente formulación lineal entera del problema ETI:

$$\begin{aligned} \text{máx} \quad & \mathbf{m}^T \mathbf{z} \\ \text{s.a} \quad & \sum_{a \in A_B(i)} z_a \geq d \quad \forall i \in I \end{aligned} \quad (3.3)$$

$$\sum_{a \in A_S(i,k)} z_a \geq 1 \quad \forall (i, k) \in \binom{I}{2}, A_S(i, k) \neq \emptyset \quad (3.4)$$

$$\begin{aligned} \sum_{a \in C \cap A_{pot}} z_a &\leq |C \cap A_{pot}| - 1 \quad \forall \text{ circuito positivo } C \text{ que contenga arcos de } A_{pot} \\ z_a &\in \{0, 1\} \quad \forall a \in A_{pot}. \end{aligned} \quad (3.5)$$

Las desigualdades (3.3) aseguran que para cada punto haya al menos d arcos de frontera en la extensión de la red de etiquetado. Estas desigualdades se puede reforzar convirtiéndolas en igualdades. Por su parte, las restricciones (3.4) evitan solapamientos entre pares de etiquetas: al menos un arco de separación debe estar presente para cada par de puntos cuyas etiquetas tengan posibilidad de solaparse. Para garantizar que existe alguna asignación factible para la red extendida solución, se añaden las restricciones (3.5). Estas últimas entrañan la dificultad del modelo, pues encontrar todos los circuitos de longitud positiva en una red es una tarea compleja. Una alternativa a la de calcular todos los circuitos a priori, es resolver el problema obviando esta familia de restricciones y, a medida que se encuentran soluciones que forman circuitos positivos, añadir las restricciones correspondientes. Para tratar de que las soluciones formen la menor cantidad de circuitos posibles, se puede minimizar el número de arcos añadidos. Esto se corresponde con tomar $m = -\mathbf{1}$.

Conviene notar que encontrar la solución del problema de optimización lineal entera anterior no supone encontrar un etiquetado factible para el conjunto de puntos \mathcal{P} . La solución

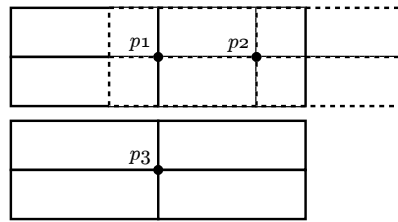


Figura 3.7: Puntos con sus regiones de etiquetado para el Ejemplo 3.4.1

correspondiente a esta formulación nos da una red de restricción para la que existe una asignación factible, la cual se traduce en la asignación de etiquetas buscada, pero no siempre sabemos cómo encontrarla. En los modelos discretos la traducción de red a asignación de etiquetas es automática: los arcos frontera determinan completamente la posición de las etiquetas. Sin embargo, en el caso continuo puede haber infinitas posibilidades a la hora de llevar a cabo esta traducción. El Ejemplo 3.4.1 nos sirve como ilustración de este hecho.

Ejemplo 3.4.1. *Supongamos que se desean etiquetar los tres puntos $p_1 = (3, 3.2)$, $p_2 = (5, 3.2)$ y $p_3 = (3, 1)$ siguiendo un modelo 4-deslizamiento con tamaño de etiqueta 1×3 . La Figura 3.7 muestra los tres puntos con las respectivas regiones donde las etiquetas son susceptibles de ser colocadas. Supongamos que como solución del problema ETI obtenemos la red de etiquetado que se muestra en la Figura 3.8. Los arcos de color azul son arcos frontera mientras que el arco rosa es de separación. Fijando nuestra atención en los arcos azules podemos deducir lo siguiente. De la red de la izquierda, sabemos que, en la solución, p_2 está en la frontera izquierda de su etiqueta y p_3 en la frontera derecha de la suya. De la red de la derecha, se deduce que p_1 está en la frontera inferior de su etiqueta. Por su parte, el arco rosa indica que las etiquetas de p_1 y p_2 están separadas horizontalmente. La Figura 3.9 muestra algunos etiquetados cumpliendo estas restricciones, que se corresponderían con distintas asignaciones de coordenadas para los nodos de la red. Sin embargo, encontrar estos etiquetados partiendo de la red extendida no es siempre una tarea sencilla (imagínese un mapa con miles de puntos para etiquetar).*

3.4.2. Formulación para ETI extendida

En su tesis, Klau presenta una alternativa a la formulación anterior, que evita las restricciones correspondientes a los circuitos positivos. Se introducen nuevas variables de decisión, correspondientes a una asignación de coordenadas, $\mathbf{c} \in \mathbb{Q}^{V_x \cup V_y}$. La formulación extendida del

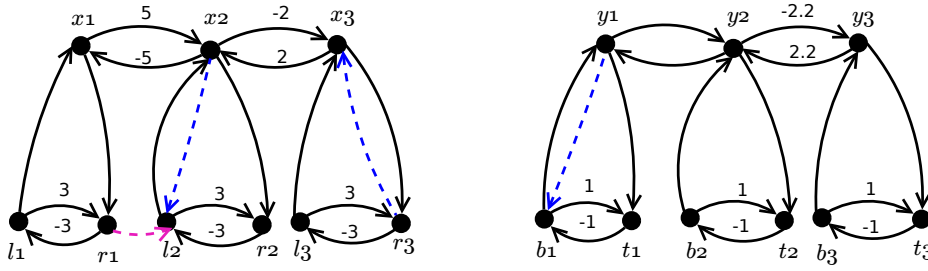


Figura 3.8: Red de etiquetado para una solución al problema del Ejemplo 3.4.1

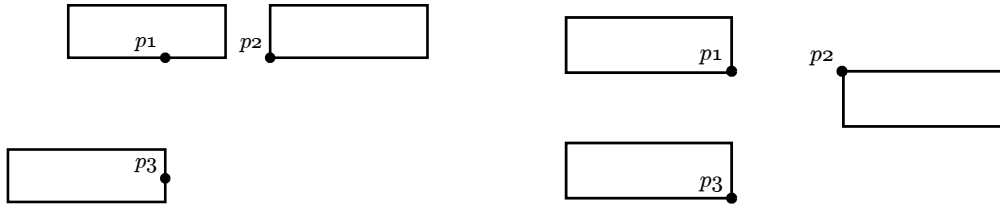


Figura 3.9: Etiquetados solución al problema del Ejemplo 3.4.1

problema ETI queda como sigue

$$\begin{aligned}
 & \text{máx} && \mathbf{m}^T \begin{pmatrix} \mathbf{z} \\ \mathbf{c} \end{pmatrix} \\
 & \text{s.a} && \sum_{a \in A_B(i)} z_a \geq d && \forall i \in I \\
 & && \sum_{a \in A_S(i,k)} z_a \geq 1 && \forall (i,k) \in \binom{I}{2}, A_S(i,k) \neq \emptyset \\
 & && c_v - c_u \geq w_x(a) && \forall a = (u,v) \in A_x && (3.6)
 \end{aligned}$$

$$c_v - c_u \geq w_y(a) \quad \forall a = (u,v) \in A_y \quad (3.7)$$

$$c_v - c_u - M_a z_a \geq -M_a \quad \forall a = (u,v) \in A_{pot} \quad (3.8)$$

$$z_a \in \{0, 1\} \quad \forall a \in A_{pot}.$$

Las restricciones (3.6) y (3.7) aseguran que la asignación de coordenadas sea factible para los arcos de la red de etiquetado, mientras que la restricción (3.8) establece la factibilidad en el caso de los arcos que forman parte de la extensión de la red. Esta última restricción, si $z_a = 1$, es decir, el arco $a = (u, v)$ está en la extensión, se traduce a la desigualdad $c_v - c_u \geq 0$. Si $z_a = 0$, para M_a suficientemente grande, la expresión (3.8) no añade restricción alguna al problema.

Esta formulación y la anterior son equivalentes, en el sentido de que ambas incluyen las mismas restricciones. La diferencia es que esta última formulación expresa la restricción de factibilidad de la asignación de coordenadas de manera directa, mientras que la anterior lo hace a través

del Teorema 3.1.1. La principal ventaja de la nueva formulación es que permite obtener directamente una asignación factible (si la hay), en lugar de asegurar que ésta existe. A continuación se establecen cotas superiores para las constantes M_a , $a \in A_{pot}$.

Lema 3.4.1. *Los siguientes valores son cotas superiores de la distancia $c_u - c_v$ y por tanto elecciones de M_a suficientemente grandes, $a = (u, v) \in A_{pot}$:*

$$M_a = \begin{cases} w_i & a = (u, v) \in A_B(i), \{u, v\} \in V_x \\ h_i & a = (u, v) \in A_B(i), \{u, v\} \in V_y \\ w_i + w_k - (\alpha_k - \alpha_i) & a = (u, v) \in A_S(i, k), \{u, v\} \in V_x \\ h_i + h_k - (\beta_k - \beta_i) & a = (u, v) \in A_S(i, k), \{u, v\} \in V_y. \end{cases}$$

Demostración. Distinguiamos dos casos:

- a es un arco frontera. Si es un arco horizontal, uno de sus extremos se corresponde con la frontera izquierda o derecha de una etiqueta y el otro con la coordenada x del punto asociado a la etiqueta. Como existe un arco de proximidad que es inverso de a , la distancia horizontal entre el punto y la frontera de la etiqueta está acotada por la anchura de la misma. De manera similar ocurre si a es vertical.
- a es un arco de separación. Supongamos que a es horizontal (para un arco vertical el razonamiento es análogo). Sin pérdida de generalidad, supongamos que $a = (r_i, l_k)$, $i, k \in I$. Si p_i y p_k están tan alejados que las regiones de etiquetado correspondientes no se solapan, tendríamos que $A_S(i, k) = \emptyset$ por lo que a no estaría considerada como arista potencial. Por tanto, las regiones de etiquetado se solapan. Si en la asignación las etiquetas de p_i y p_k están separadas horizontalmente, podría ocurrir que $c_u - c_v \approx w_i + w_k > M_a$, como muestra la Figura 3.10. En este caso, establecer el valor de la cota $M_a = w_i + w_k - (\alpha_k - \alpha_i)$ obliga a añadir el arco de separación correspondiente, es decir, $z_a = 1$, para verificar la restricción (3.8). En otro caso, si en la asignación las etiquetas de p_i y p_k no están horizontalmente separadas, tendremos una situación como la ilustrada por la Figura 3.11. Aquí puede comprobarse que en el peor de los casos $c_u - c_v = c_{r_i} - c_{l_k} = w_i + w_k - (\alpha_k - \alpha_i)$.

□

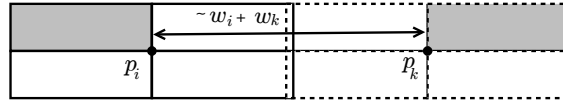


Figura 3.10: Ejemplo de dos etiquetas separadas horizontalmente.

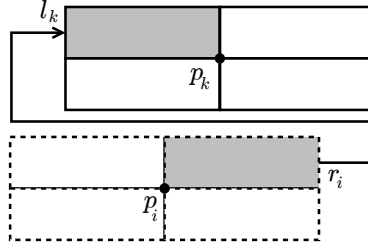


Figura 3.11: Ejemplo de dos etiquetas no separadas horizontalmente.

3.4.3. Formulación para MNE

El problema de maximización de etiquetas difiere del problema de etiquetado puro en que se debe encontrar un subconjunto de puntos que serán etiquetados. En el problema puro, debe darse un etiquetado de los puntos o un mensaje diciendo que este etiquetado no es posible, mientras que en el problema de maximización siempre existe solución, no etiquetar ningún punto. Para desarrollar la formulación del problema MNE, se debe modelar la elección de los puntos que se van a etiquetar, $I' \subseteq I = \{1, \dots, n\}$. Se definen las variables $s \in \{0, 1\}^n$:

$$s_i = \begin{cases} 1 & i \in I', \\ 0 & \text{en otro caso.} \end{cases}$$

La formulación inicial de este problema parte de la formulación del problema ETI, adaptando las restricciones de los arcos frontera y solapamiento para que solo afecten a los puntos correspondientes a I' :

$$\begin{aligned} \text{máx} \quad & \sum_{i \in I} s_i \\ \text{s.a} \quad & \sum_{a \in A_B(i)} z_a \geq d \cdot s_i \quad \forall i \in I \end{aligned} \quad (3.9)$$

$$\sum_{a \in A_S(i,k)} z_a - s_i - s_k \geq -1 \quad \forall (i, k) \in \binom{I}{2}, A_S(i, k) \neq \emptyset \quad (3.10)$$

$$\sum_{a \in C \cap A_{pot}} z_a \leq |C \cap A_{pot}| - 1 \quad \forall \text{ circuito positivo } C \text{ que contenga arcos de } A_{pot}$$

$$s_i \in \{0, 1\} \quad \forall i \in I.$$

$$z_a \in \{0, 1\} \quad \forall a \in A_{pot}.$$

Las restricciones (3.9) no añaden información si $s_i = 0$, es decir, $i \notin I'$, y son equivalentes a (3.3) si $i \in I$. Por otro lado, las desigualdades (3.10) actúan como (3.4) si $s_i = s_k = 1$. En otro caso, no añaden ninguna restricción al modelo.

En este punto, recordamos que las restricciones (3.3) del problema ETI pueden reforzarse para convertirse en igualdades:

$$\sum_{a \in A_B(i)} z_a = d \quad \forall i \in I.$$

En el problema MNE ocurre lo mismo, podemos reemplazar (3.9) por:

$$\sum_{a \in A_B(i)} z_a = d \cdot s_i \quad \forall i \in I.$$

Esto nos permite sustituir s_i por $\sum_{a \in A_B(i)} \frac{z_a}{d}$ en la formulación. El número de arcos frontera que hay en la extensión de la red dividido por d nos da el número de puntos etiquetados. Por tanto, el objetivo es ahora maximizar el número de arcos frontera. Hay que tener en cuenta dos consideraciones a este respecto:

- Consideremos el caso de una asignación factible para un modelo con deslizamiento. Recordamos que en este tipo de modelos un punto debe pertenecer a la frontera de su etiqueta, restricción que se modela a través de arcos frontera. Supongamos que en dicha asignación un punto $\bar{i} \in I$ coincide con una de las esquinas de su etiqueta. Si esto ocurre, debe ser posible añadir los dos arcos frontera correspondientes en la red de etiquetado. En la formulación anterior, como el objetivo es maximizar el número de arcos frontera, se añadirían estos dos arcos frontera a la extensión. Por lo tanto, este punto \bar{i} contaría por dos en la función objetivo, que es la suma $\sum_{i \in I} \sum_{a \in A_B(i)} \frac{z_a}{d}$. Es decir, el valor objetivo asociado a esta solución sería el mismo si en lugar de etiquetar \bar{i} haciéndolo coincidir con la esquina de su etiqueta, etiquetamos dos puntos sin que coincidan con ninguna esquina de sus respectivas etiquetas. Para evitar esto acotamos el número de arcos frontera según el modelo añadiendo la siguiente restricción:

$$\sum_{a \in A_B(i)} z_a \leq d \quad \forall i \in I.$$

Si el modelo es discreto estas restricciones no son necesarias, pues en una solución factible nunca aparecen más de dos arcos frontera para un mismo punto (esto induciría un circuito positivo en la red extendida resultante).

- En los modelos discretos, no deben permitirse etiquetados en los que un punto no coincida con alguna de las esquinas de su etiqueta. Para conseguirlo, se evita que para un punto solo un arco frontera aparezca en la extensión. Para cada punto, se añade una restricción que asegura que el número de arcos frontera verticales coincide con el número de arcos frontera horizontales. Para el caso del modelo *4-posición* se incluye:

$$z_{(r_i, x_i)} + z_{(x_i, l_i)} = z_{(t_i, y_i)} + z_{(y_i, b_i)} \quad \forall i \in I.$$

Para el modelo *2-posición* ($t_i, y_i \notin A_B(i)$), por lo que la restricción es:

$$z_{(r_i, x_i)} + z_{(x_i, l_i)} = z_{(y_i, b_i)} \quad \forall i \in I.$$

Finalmente, para el modelo *1-posición* tendríamos:

$$z_{(x_i, l_i)} = z_{(y_i, b_i)} \quad \forall i \in I.$$

La formulación del problema MNE, introduciendo los cambios anteriores, quedaría como sigue:

$$\begin{aligned}
& \text{máx} && \sum_{i \in I} \sum_{a \in A_B(i)} \frac{z_a}{d} \\
& \text{s.a} && \sum_{a \in A_S(i, k)} z_a - \sum_{a \in A_B(i)} \frac{z_a}{d} - \sum_{a \in A_B(k)} \frac{z_a}{d} \geq -1 \quad \forall (i, k) \in \binom{I}{2}, A_S(i, k) \neq \emptyset \\
& && \sum_{a \in A_B(i)} z_a \leq d \\
& && \sum_{a \in C \cap A_{pot}} z_a \leq |C \cap A_{pot}| - 1 \quad \forall \text{ circuito positivo } C \text{ que contenga arcos de } A_{pot} \\
& && z_{(r_i, x_i)} + z_{(x_i, l_i)} = z_{(t_i, y_i)} + z_{(y_i, b_i)} \quad \forall i \in I \text{ si } d = 2. \\
& && z_a \in \{0, 1\} \quad \forall a \in A_{pot}.
\end{aligned}$$

Capítulo 4

Una nueva aproximación

Los modelos de etiquetado de mapas existentes en la literatura, de un modo u otro, se centran en los solapamientos como medida de la bondad de una determinada distribución de las etiquetas. Ejemplo de ello son los modelos de Verweij, 2000 y Klau, 2001, presentados en los capítulos anteriores, así como los propuestos por Mauri et al., 2010, Zoraster, 1990 y Ribeiro and Lorena, 2008, entre otros.

En este último capítulo se presenta una nueva aproximación, propuesta por Marín and Pelegrín, 2016, que se basa en la ambigüedad de los etiquetados para decidir cuándo uno es mejor que otro. Cuando hablamos de ambigüedad nos referimos a una dificultad a la hora de identificar los puntos descritos por cada una de las etiquetas. Es posible que, en regiones en las que los puntos están muy concentrados, puedan confundirse las asociaciones punto-etiqueta. Un ejemplo de esto es el mostrado por la Figura 4.1. El etiquetado que aparece la parte superior de la figura es claramente más ambiguo que el que se encuentra en la parte inferior. La razón es que en el primer caso las esquinas de las etiquetas están muy juntas entre sí. Los modelos que para evaluar las soluciones únicamente tienen en cuenta el número de solapamientos, encontrarían los dos etiquetados mostrados en la Figura 4.1 igual de buenos.

En este capítulo, presentaremos cuatro nuevos modelos, agrupados por parejas. Los dos primeros se basan en establecer clases de ambigüedad junto con un peso para cada una de las clases. Estos modelos minimizan el número de etiquetas que pertenecen a cada clase, donde la preferencia de minimización entre varias clases viene determinada por los pesos de las mismas. Los dos últimos modelos siguen una idea parecida, pero resultan más flexibles. Estos modelos ordenan las etiquetas de menor a mayor ambigüedad siendo el objetivo que las últimas etique-

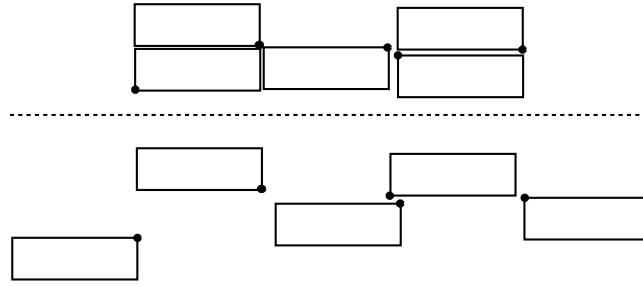


Figura 4.1: Dos etiquetados distintos para un mismo conjunto de puntos.

tas de la ordenación sean lo menos ambiguas posible. Cada pareja de modelos se corresponde a una distinción del caso en que todos los puntos han de ser etiquetados, independientemente de que esto implique que haya solapamientos, y el caso en que se permiten puntos sin etiquetar (si los solapamientos no puedan evitarse de otro modo).

Supondremos que las etiquetas son rectángulos, alguna de sus esquinas coincidiendo con el punto al que etiquetan. De otro modo, sería necesario modificar ligeramente los modelos propuestos, aunque se podrían adaptar sin grandes dificultades. Para medir la ambigüedad de una etiqueta se tiene en cuenta lo cerca que están sus esquinas del conjunto de puntos que intervienen en el modelo. En concreto, la ambigüedad de una etiqueta $\ell \in L_i$ viene dada por el mínimo de las distancias desde cualquiera de las esquinas de ℓ (excepto la esquina p_i) hasta cualquiera de los puntos en $\mathcal{P} \setminus \{p_i\}$. Cuanto más corta sea esta distancia mayor será la ambigüedad producida por la etiqueta. La razón de no tener en cuenta la esquina p_i es que las distancias de otros puntos a este no son representativas puesto que, como los puntos del modelo vienen dados de antemano, nada puede hacerse por evitar que dos de estos puntos sean cercanos.

Consideraremos $(\mathcal{P} = \{p_1, \dots, p_n\}, L = \cup_{i \in I} L_i)$ una instancia del problema del etiquetado, donde $I = \{1, \dots, n\}$ y L_i es el conjunto de posibles etiquetas para cada punto p_i , $i \in I$.

4.1. Modelos basados en clases de ambigüedad

En estos modelos, cada etiqueta se incluye en una clase dependiendo de la ambigüedad que suponga utilizarla. El conjunto de clases será $R = \{1, \dots, \alpha\}$, donde cada clase $r \in R$ se refiere a las etiquetas $\ell \in L$ tales que $d_\ell \in I_r = [C_{r-1}, C_r)$. Los valores $C_0 := 0 < C_1 < \dots < C_\alpha := \infty$ establecen los límites de las clases, de forma que I_1 es la clase de ambigüedad extrema, I_2 representa gran ambigüedad, etc. Supongamos ahora que tenemos un etiquetado de los puntos, $\rho : I \rightarrow L$. Cada punto p_i , $i \in I$, tendrá asignada una etiqueta $\rho(i)$, que para algún $r \in R$

cumplirá $C_{r-1} \leq d_{\rho(i)} < C_r$. Diremos entonces que la etiqueta $\rho(i)$, o simplemente el punto p_i , pertenece a la clase r .

La ambigüedad global del etiquetado ρ se obtiene a partir del número de etiquetas en cada clase. Denotemos con n_r el número de etiquetas en la clase r , $r \in R$. La idea principal consiste en dar un peso relativo a cada valor de ambigüedad $d_{\rho(i)}$, $i \in I$, dependiendo de la clase a la que pertenezca. Estos pesos, que llamaremos μ_r , $\mu_r \geq 0$ para cada $r \in R$, pueden variar de acuerdo al gusto del decisor. Cada etiqueta en la clase r aportará un valor μ_r a la función objetivo. Es decir, cada clase $r \in R$ sumará un total de $n_r \mu_r$ al objetivo. Como el objetivo es maximizar, el vector $\mu = (\mu_1, \dots, \mu_\alpha)$ deberá ser creciente, de manera que una etiqueta aporte más valor a la función objetivo cuanto mayor sea la clase a la que pertenece (y por tanto menor ambigüedad produzca). Por ejemplo, si el decisor elige $\mu_1 = 0$ y el resto de μ 's toman el valor 1, el modelo tratará de evitar etiquetas extremadamente ambiguas y no prestará atención al resto de clases. Si elige $\mu = (\mu_1, \dots, \mu_\rho) = (0, \frac{M}{2^{\alpha-2}}, \dots, \frac{M}{2}, M)$, en caso de haber varias posibilidades para etiquetar un punto, se elegirá la etiqueta que pertenezca a una clase de ambigüedad lo mayor posible.

Consideramos las siguientes variables:

$$x_\ell = 1 \text{ si } \ell = \rho(i) \text{ para algún } i \in I, \ell \in L,$$

$$y_i = 1 \text{ si el punto } i \in I \text{ es etiquetado.}$$

A continuación se formula el modelo anterior, en el caso en que se permite que queden puntos sin etiquetar:

$$\begin{aligned} \text{(MP1) máx} \quad & \sum_{i \in I} M y_i + \sum_{r=1}^{\rho-1} \sum_{\substack{\ell \in L: \\ C_r \leq d_\ell < C_{r+1}}} \mu_r x_\ell \\ \text{s.a} \quad & \sum_{\ell \in L_i} x_\ell = y_i \quad \forall i \in I \end{aligned} \quad (4.1)$$

$$x_\ell + \sum_{\substack{j \in L_i: \\ j \cap \ell \neq \emptyset}} x_j \leq 1 \quad \forall i \in I, \ell \notin L_i \quad (4.2)$$

$$y_i \in \{0, 1\} \quad \forall i \in I$$

$$x_\ell \in \{0, 1\} \quad \forall \ell \in L.$$

Las restricciones recogidas en (4.1) hacen que y_i sea igual a 1 si alguna de las variables x_ℓ lo es, $\ell \in L_i$. Las restricciones (4.2) evitan que existan solapamientos en la solución. Distinguiamos dos partes en la función objetivo. La primera, $\sum_{i \in I} M y_i$, garantiza la maximización del número

de puntos etiquetados. Si M es un valor suficientemente grande, este será el objetivo prioritario. La segunda parte trata que, en caso de igualdad en el número de puntos etiquetados, se escoja la solución menos ambigua.

A continuación se presenta una modificación de la formulación anterior, que puede ser de utilidad cuando todos los puntos del modelo han de ser etiquetados:

$$\begin{aligned}
(\text{MP1}') \quad & \text{máx} \quad \sum_{i \in I} -My'_i + \sum_{r=1}^{\rho-1} \sum_{\substack{\ell \in L: \\ C_r \leq d_\ell < C_{r+1}}} \mu_r x_\ell \\
\text{s.a} \quad & \sum_{\ell \in L_i} x_\ell = 1 \quad \forall i \in I \\
& x_\ell + \sum_{\substack{j \in L_i: \\ j \cap \ell \neq \emptyset}} x_j \leq 1 + y'_i \quad \forall i \in I, \ell \notin L_i \quad (4.3) \\
& y'_i \geq 0 \quad \forall i \in I \\
& x_\ell \in \{0, 1\} \quad \forall \ell \in L.
\end{aligned}$$

Aquí, las variables y'_i se usan para limitar el número de etiquetas que se solapan. Aunque no se definen como variables binarias, estas variables toman valor 0 o 1 en la solución óptima. Para comprobarlo, basta con observar que $\sum_{\substack{j \in L_i: \\ j \cap \ell \neq \emptyset}} x_j$ vale como mucho 1, por lo que el lado izquierdo de la desigualdad (4.3) vale como mucho 2. Como el primer término de la función objetivo a maximizar es $-My'_i$, el lado derecho de esta desigualdad será lo más ajustado posible, valiendo las variables positivas y'_i 0 o 1. De esta forma, estas variables contabilizan el número de etiquetas que se solapan con otras, es decir, $y'_i = 1$ si y sólo si $\rho(i)$ se solapa con $\rho(i')$ para algún $i' \in I \setminus \{i\}$.

4.2. Modelos con ordenación

La bondad de los modelos anteriores depende fuertemente de la elección de los límites de las clases $C_1, \dots, C_{\alpha-1}$. Una vez que tenemos una instancia concreta de un mapa, un buen criterio para definir estos valores debería estar relacionado con las distancias existentes entre todas las etiquetas que intervienen. De otra forma, podría ocurrir que todas las etiquetas pertenecieran a una misma clase de ambigüedad, quedando el modelo totalmente invalidado.

En esta sección se presenta una clase de modelo más flexible, de forma que no es necesario su ajuste en función de la instancia a resolver. Recordamos que la ambigüedad es producida por pequeños valores de las distancias d_ℓ , $\ell \in L$. Los modelos que vamos a presentar a conti-

nuación no tienen en cuenta las etiquetas que aportan poca ambigüedad a la solución. Así, se establece un valor umbral T para estas distancias por encima del cual se considera que no hay ambigüedad ninguna. Para cada $i \in I$, se define el conjunto de etiquetas ambiguas asociadas al punto p_i como $L_i^T := \{\ell \in L_i : d_\ell \leq T\}$.

De nuevo, sea $\rho : I \rightarrow L$ un etiquetado de los puntos. Algunas de las etiquetas $\rho(i)$, $i \in I$, serán ambiguas, es decir, $\rho(i) \in L_i^T$, y otras no. Sin pérdida de generalidad, supongamos que $\rho(1), \dots, \rho(t)$, $t \leq n$, son todas las etiquetas cuyas distancias asociadas están por debajo del umbral T . En lugar de agrupar estas etiquetas por clases, consideremos una permutación de las mismas σ para la cual se satisface:

$$d_{\rho(\sigma(1))} \geq d_{\rho(\sigma(2))} \geq \dots \geq d_{\rho(\sigma(t))},$$

o equivalentemente:

$$d_{\rho(\sigma(1))} - T \geq d_{\rho(\sigma(2))} - T \geq \dots \geq d_{\rho(\sigma(t))} - T.$$

La función objetivo se obtiene en parte multiplicando estos últimos valores ordenados por un conjunto de parámetros $\lambda_i \geq 0$, es decir, se desea añadir al objetivo el término no lineal:

$$\lambda_1(d_{\rho(\sigma(1))} - T) + \lambda_2(d_{\rho(\sigma(2))} - T) + \dots + \lambda_t(d_{\rho(\sigma(t))} - T). \quad (4.4)$$

Al igual que en los modelos de la sección anterior, estos parámetros pueden variar en función del gusto del decisor. Como el objetivo es maximizar, λ_i , $i = 1, \dots, t$ debe ser una sucesión creciente. Así, cuanto más ambigua sea una etiqueta $\rho(i)$ (es decir, cuanto mayor sea el puesto ocupado por $d_{\rho(i)}$ en la ordenación anterior), mayor será su contribución para que la función objetivo crezca. De esta forma, se consigue que $d_{\rho(i)}$ sea lo más grande posible en el óptimo. Finalmente, se han considerado los términos negativos $d_{\rho(\sigma(i))} - T$, de manera que si una etiqueta $\rho(i)$ no es ambigua (es decir, $d_{\rho(i)} \geq T$), el valor que aporta esta a la función objetivo es mayor que el que aportaría cualquier etiqueta ambigua (el modelo descarta las etiquetas no ambiguas, lo que equivale a sumar 0 a la función objetivo).

Para transformar las ideas expuestas en un modelo de optimización combinatoria, primero es necesario ordenar los valores d_ℓ de todas las etiquetas presentes en $L^T := \cup_{i \in I} L_i^T$ en orden

creciente (descartando empates) como:

$$(T \geq) d'_1 > d'_2 > \cdots > d'_K > 0.$$

Para elaborar el modelo matemático, extendemos la permutación σ de forma que actúe sobre todos los puntos del modelo, es decir, de manera que se cumpla:

$$d_{\rho(\sigma(1))} \geq d_{\rho(\sigma(2))} \geq \cdots \geq d_{\rho(\sigma(n))}.$$

Asimismo, también supondremos que el vector lambda tiene n componentes, λ_i , $i = 1, \dots, n$. Estas extensiones son necesarias para la correcta definición de las variables siguientes. Introducimos un conjunto de variables binarias, definidas para $i \in I$, $k \in \{1, \dots, K\}$:

$$s_{ik} = 1 \text{ si } d_{\rho(\sigma(i))} \leq d'_k.$$

Usando las anteriores junto con las variables \mathbf{x} e \mathbf{y} de las formulaciones de la sección anterior, la nueva formulación viene dada por:

$$\begin{aligned}
 \text{(MP2) máx} \quad & \sum_{i \in I} M y_i + \sum_{i=1}^n \lambda_i \left(\sum_{k=1}^{K-1} (d'_k - T)(s_{ik} - s_{i,k+1}) + (d'_K - T)s_{iK} \right) \\
 \text{s.a} \quad & \sum_{\ell \in L_i} x_\ell = y_i \quad \forall i \in I \quad (4.5) \\
 & x_\ell + \sum_{\substack{j \in L_i: \\ j \cap \ell \neq \emptyset}} x_j \leq 1 \quad \forall i \in I, \ell \notin L_i \quad (4.6) \\
 & \sum_{i \in I} \sum_{\substack{\ell \in L_i^T: \\ d_\ell \leq d'_k}} x_\ell = \sum_{i=1}^n s_{ik} \quad \forall k = 1, \dots, K \quad (4.7) \\
 & s_{ik} \leq s_{i+1,k} \quad \forall i = 1, \dots, n-1, \\
 & \quad \quad \quad \forall k = 1, \dots, K \quad (4.8) \\
 & y_i \in \{0, 1\} \quad \forall i \in I \\
 & x_\ell \in \{0, 1\} \quad \forall \ell \in L \\
 & s_{ik} \in \{0, 1\} \quad \forall i \in I, k = 1, \dots, K.
 \end{aligned}$$

De forma similar a las formulaciones (MP1) y (MP1'), las restricciones (4.5) y (4.6) controlan las etiquetas utilizadas. Las restricciones (4.7) y (4.8) hacen que las variables \mathbf{s} resulten en una matriz cuyas filas están ordenadas por el número de ceros que contienen. Cada fila se corresponde con una etiqueta y nos da la información de la ambigüedad de la misma. Debido a la restricción (4.7), al comienzo de cada fila habrá una cantidad determinada de

unos y el resto serán ceros. Para cada $i \in I$, el mayor subíndice $k \in \{1, \dots, K\}$ para el cual $s_{ik} = 1$ verificará que $d_{\rho(i)} = d'_k$. Si $s_{ik} = 0$ para todo $k \in \{1, \dots, K\}$ entonces $d_{\rho(i)} > d'_k \forall k \in \{1, \dots, K\}$, en particular $d_{\rho(i)} > T$ (es inmediato ver que la implicación contraria también se cumple). En este último caso tendremos que el término $\sum_{k=1}^{K-1} d'_k (s_{ik} - s_{i,k+1}) + d'_K s_{iK}$ de la función objetivo es cero. Por otro lado, si $s_{ik} = 1$ para algún $k \in \{1, \dots, K\}$ entonces $\sum_{k=1}^{K-1} d'_k (s_{ik} - s_{i,k+1}) + d'_K s_{iK} = d'_{k_{max}}$ donde k_{max} es el máximo índice para el cual $s_{ik_{max}} = 1$. Así, la expresión de la función objetivo $\sum_{i=1}^n \lambda_i (\sum_{k=1}^{K-1} d'_k (s_{ik} - s_{i,k+1}) + d'_K s_{iK})$ se corresponde correctamente con (4.4). El otro término que forma la función objetivo, $\sum_{i \in I} M y_i$, garantiza que el número de puntos etiquetados se maximiza.

La formulación (MP2) permite que queden puntos sin etiquetar. A continuación, presentamos una modificación para el caso en que todos los puntos deben ser etiquetados:

$$\begin{aligned}
 \text{(MP2')} \quad \text{máx} \quad & \sum_{i \in I} -M y'_i + \sum_{i=1}^n \lambda_i \left(\sum_{k=1}^{K-1} (d'_k - T)(s_{ik} - s_{i,k+1}) + (d'_K - T)s_{iK} \right) \\
 \text{s.a} \quad & \sum_{\ell \in L_i} x_\ell = 1 && \forall i \in I \\
 & x_\ell + \sum_{\substack{j \in L_i: \\ j \cap \ell \neq \emptyset}} x_j \leq 1 + y'_i && \forall i \in I, \ell \notin L_i \\
 & \sum_{i \in I} \sum_{\substack{\ell \in L_i^T: \\ d_\ell \leq d'_k}} x_\ell = \sum_{i=1}^n s_{ik} && \forall k = 1, \dots, K \\
 & s_{ik} \leq s_{i+1,k} && \forall i = 1, \dots, n-1, \\
 & && \forall k = 1, \dots, K \\
 & y'_i \geq 0 && \forall i \in I \\
 & x_\ell \in \{0, 1\} && \forall \ell \in L \\
 & s_{ik} \in \{0, 1\} && \forall i \in I, k = 1, \dots, K.
 \end{aligned}$$

De nuevo, y'_i se emplea para limitar el número de solapamientos y, aunque no está definida como variables binaria, tomará el valor 0 o 1 en el óptimo.

Nótese que, si se incluyeran las distancias que son mayores o iguales que T en el objetivo sin tener en cuenta el orden, obtendríamos una solución que minimizaría la ambigüedad en general. Esto implicaría que varias etiquetas extremadamente ambiguas podrían compensar una gran cantidad de etiquetas no ambiguas, cuando en muchos casos es preferible una distribución de la ambigüedad más equilibrada. El modelo con ordenación que proponemos nos permite controlar este efecto. Además, su flexibilidad permite expresar objetivos bastante específicos.

Por ejemplo, si se permiten valores lambdas negativos, sería posible forzar que todas las etiquetas fueran ambiguas en una medida similar tomando $\lambda = (-1, 0, \dots, 0, 1)$. Podemos encontrar modelos previos en la literatura con objetivos ordenados, que han sido utilizados con éxito en el campo de la localización discreta, como Marín et al., 2009 y Marín et al., 2010.

4.3. Algunas propiedades

Como los valores d'_j están ordenados de manera decreciente ($d'_1 > d'_2 > \dots > d'_K$), si $d_{j_{\sigma(i)}} < d'_k$ entonces también se cumple que $d_{j_{\sigma(i)}} < d'_i$ para todo $i = 1, \dots, k - 1$. Por lo tanto, las siguientes son desigualdades válidas para las formulaciones (MP2) y (MP2'):

$$s_{ik} \geq s_{i,k+1} \quad \forall i = 1, \dots, n, \quad \forall k = 1, \dots, K - 1.$$

Por otro lado, las variables y_i de los modelos (MP1) y (MP2) pueden omitirse. En el objetivo, $\sum_{i \in I} My_i$ puede reemplazarse por $\sum_{i \in I} \sum_{\ell \in L_i} Mx_\ell$. Para completar el cambio, las restricciones (4.5) y (4.1) se transformarían en: $\sum_{\ell \in L_i} x_\ell \leq 1$. Además, en estos modelos pueden suprimirse las etiquetas que tocan a un punto distinto al que describen, ya que estas siempre darán lugar a solapamientos.

Con el fin de hacer el proceso del etiquetado más simple, el problema puede dividirse en varios subproblemas independientes. Consideremos, por ejemplo, un mapa describiendo las ciudades más importantes de un determinado país. La etiqueta que describe una ciudad dada puede no afectar al resto. Este podría ser el caso en las zonas menos pobladas que separan varios núcleos urbanos, que de manera natural evitarían potenciales conflictos en el etiquetado. La idea consiste en identificar estas áreas en el caso general para resolver así varios subproblemas de menor tamaño en lugar de la instancia completa. Una región independiente del problema será un conjunto de puntos verificando:

1. Para cada punto p_i de la región, existe otro punto $p_{i'}$ en la misma región tal que al menos dos de las potenciales etiquetas asociadas a los respectivos puntos se solapan, es decir, $L_i \cap L_{i'} \neq \emptyset$.
2. Si los puntos p_i y $p_{i'}$ pertenecen a distintas regiones entonces $L_i \cap L_{i'} = \emptyset$.

En este contexto, conviene recordar que todos los modelos propuestos aúnan dos objetivos: evitar o minimizar los solapamientos y reducir la ambigüedad. Respecto a los solapamientos, las

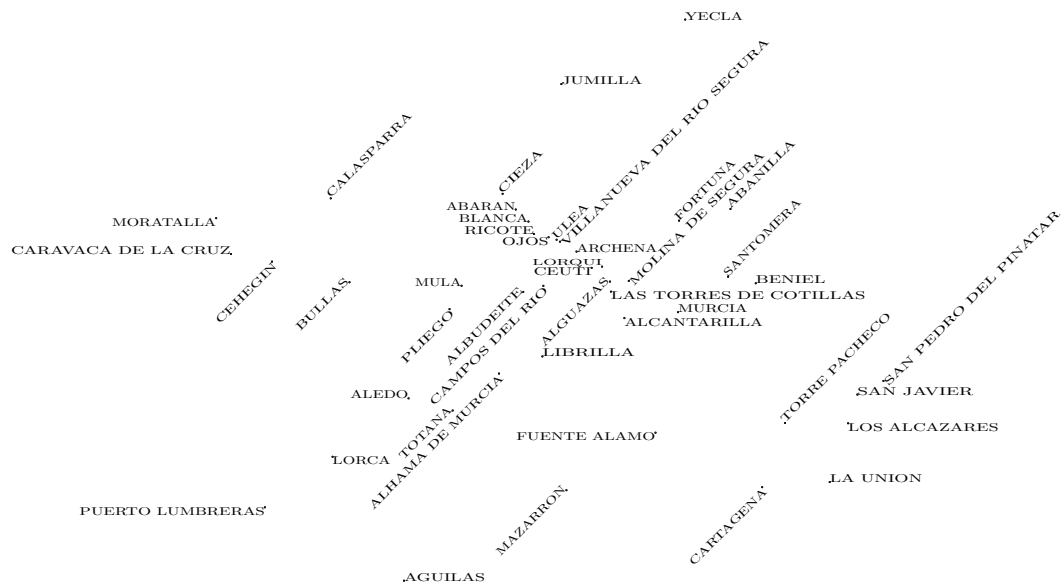
dos restricciones anteriores dividen el mapa en regiones totalmente independientes. En cambio, para la ambigüedad no ocurre lo mismo. La ambigüedad de una etiqueta dada puede estar producida por la cercanía a otra etiqueta correspondiente a un punto de una región del mapa distinta. Sin embargo, esto no supone dificultad extra, dado que las distancias d_ℓ , $\ell \in L$, se calculan a partir de los datos del problema. Estas distancias pueden ser preprocesadas antes de llevar a cabo la división en subproblemas, y utilizadas posteriormente en cada uno de los modelos resultantes.

Finalmente, sería interesante tener un índice normalizado entre 0 y 1 que nos indicara la ambigüedad del etiquetado. En todos los modelos propuestos, la función objetivo consta de dos partes, una de las cuales se refiere a la ambigüedad. Por tanto, bastará con extraer esta parte y hacer algunas operaciones de normalización, teniendo en cuenta que la parte extraída es mayor cuanto menor es la ambigüedad. En concreto, se realizan los siguientes cálculos:

- Obtener un valor de ambigüedad a como resultado de sustraer/ añadir al valor objetivo la cantidad M tantas veces como puntos etiquetados/ número de solapamientos, dependiendo del modelo.
- Para cada $i \in I$, obtener $d_{max_i} = \max\{d_j : j \in L_i\}$ y $d_{min_i} = \min\{d_j : j \in L_i\}$.
- Obtener a_{max} y a_{min} como los valores de ambigüedad correspondientes a la función objetivo del problema en los casos en que los puntos se etiquetan con las etiquetas correspondientes a d_{max_i} y d_{min_i} respectivamente, $i \in I$.
- El valor normalizado se calcula como:

$$\left(1 - \frac{a}{a_{max}}\right) \left(\frac{a_{max}}{a_{max} - a_{min}}\right).$$

La Figura 4.2 muestra dos etiquetados distintos de los 45 municipios de la Región de Murcia, obtenidos aplicando el modelo (MP1). Esta figura pretende ilustrar la importancia de una buena elección de los límites $C_1, \dots, C_{\alpha-1}$ de las clases de ambigüedad. Aunque en los dos casos mostrados se consigue etiquetar todos los municipios, vemos que el etiquetado de la parte superior resulta menos ambiguo que el situado en la parte inferior. En el primero, los límites $C_1, \dots, C_{\alpha-1}$ se han elegido en relación a las distancias entre los municipios, de manera que se haga una verdadera clasificación de las etiquetas. En el segundo, el límite C_1 es grande en relación a las distancias que aparecen en el mapa, de manera que todas las etiquetas pertenecen a la clase $[C_0, C_1)$.



(a) Etiquetas en distintas clases de ambigüedad.



(b) Todas las etiquetas en la misma clase de ambigüedad.

Figura 4.2: Etiquetado de los 45 municipios de la Región de Murcia.



Figura 4.3: Etiquetado de los municipios españoles con etiqueta pequeña.

Por su parte, las Figuras 4.3, 4.4 y 4.5 muestran distintos etiquetados de los municipios españoles con más de 100000 habitantes (58 en total), obtenidos a través de la formulación (MP2). Vemos que, conforme aumenta el tamaño de etiqueta elegido, se hace más difícil etiquetar todos los puntos sin que haya solapamientos. Así, en el primer ejemplo todos los puntos son etiquetados, mientras que en el segundo caso queda un punto sin etiquetar y en el tercero tres (señalados en rojo).



Figura 4.4: Etiquetado de los municipios españoles con etiqueta mediana.



Figura 4.5: Etiquetado de los municipios españoles con etiqueta grande.

Capítulo 5

Conclusiones y trabajo futuro

Este capítulo resume las conclusiones generales del trabajo realizado y presenta una pequeña muestra de las extensiones futuras que se podrían llevar a cabo en el ámbito del etiquetado de mapas.

5.1. Conclusiones

Podemos extraer las siguientes conclusiones a partir de los modelos estudiados:

- El problema de etiquetar el máximo número de puntos sin solapamientos (MNE) es equivalente al problema de encontrar el empaquetamiento de mayor cardinal en un grafo. En su tesis Verweij realiza esta identificación y propone un algoritmo de ramificación y corte para resolver el problema del empaquetamiento. La originalidad de este algoritmo reside en las estrategias para fijar variables y el procedimiento de separación, basado en añadir desigualdades correspondientes a agujeros impares.
- Los problemas de etiquetar un conjunto de puntos sin solapamiento (ETI), maximizar el número de puntos etiquetados sin solapamiento (MNE) y minimizar el número de solapamientos (MNS), son equivalentes al de encontrar una red cumpliendo unas determinadas propiedades que no contenga circuitos de longitud positiva. En su tesis Klau desarrolla esta caracterización y presenta algunas formulaciones de estos problemas basadas en ella.
- Tener en cuenta la ambigüedad de los etiquetados puede ser útil para mejorar los modelos existentes. A la hora de interpretar un mapa no sólo es importante que las etiquetas sean legibles en el sentido de que no se solapen sino también que sea fácil identificar a qué punto se refiere cada etiqueta.

5.2. Trabajo futuro

No existe una vasta literatura referente al etiquetado de mapas abordado como un problema de optimización. Por ello, son muchas las aproximaciones que pueden llevarse a cabo y que están aún sin explorar. Por ejemplo, la distribución de los puntos del modelo, en lugar de ser un dato del problema puede considerarse como una variable más de decisión. Esto permitiría etiquetar elementos como curvas o regiones bidimensionales del plano con una mayor flexibilidad. Por otro lado, con la proliferación de aplicaciones de visualización en interfaces gráficas, tiene sentido la extensión de los modelos clásicos, pensados para el problema del mapa cartográfico, a modelos en tres dimensiones. Otra aproximación al problema del etiquetado que no ha sido abordada en este trabajo sería la de considerar el tamaño de las etiquetas variable. Un modelo podría consistir en encontrar el máximo factor de escala, de manera que al reducir las etiquetas por dicho factor no existan solapamientos. Otra opción podría consistir en aplicar factores de escala distintos a las etiquetas, de forma que estas puedan aumentarse o reducirse de manera individual, para formar un etiquetado lo más legible posible.

Bibliografía

- H.L. Bodlaender. Treewidth: Algorithmic techniques and results. *Lecture Notes in Computer Science*, 1295:19–36, 1997.
- W.J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. John Wiley & Sons, 1998.
- E. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1: 269–271, 1959.
- B. De Fluiter. *Algorithms for Graphs of Small Treewidth*. PhD thesis, Department of Computer Science, Universidad de Utrecht, 1997.
- M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin, 1988.
- G.W. Klau. *A Combinatorial Approach to Orthogonal Placement Problems*. PhD thesis, Universidad de Saarlandes, Alemania, 2001.
- A. Marín and M. Pelegrín. Towards unambiguous map labeling: An integer programming approach. 2016.
- A. Marín, S. Nickel, J. Puerto, and S. Velten. A flexible model and efficient solution strategies for discrete location problems. *Discrete Applied Mathematics*, 157:1128–1145, 2009.
- A. Marín, S. Nickel, and S. Velten. An extended covering model for flexible discrete and equity location problems. *Mathematical Methods of Operations Research*, 71:125–163, 2010.
- G.R. Mauri, G.M. Ribeiro, and L.A.N. Lorena. A new mathematical model and a lagrangian decomposition for point-feature cartographic label placement problem. *Computers & Operations Research*, 37:2164–2172, 2010.

-
- G.L. Nemhauser and G. Sigismondi. A strong cutting plane/branch-and-bound algorithm for node packing. *Journal of the Operations Research Society*, 43:443–457, 1992.
- M.W. Padberg. On the facial structure of set packing polyhedra. *Mathematical Programming*, 5:199–215, 1973.
- G.M. Ribeiro and L.A.N. Lorena. Lagrangean relaxation with clusters for the point-feature cartographic label placement problem. *Computers & Operations Research*, 35:2129–2140, 2008.
- T. Strijk, B. Verweij, and K. Aardal. Algorithms for maximum independent set applied to map labelling. Technical report, Universidad de Utrecht, 2000.
- A. M. Verweij. *Selected Applications of Integer Programming: A Computational Study*. PhD thesis, Universidad de Utrecht, 2000.
- L.A. Wolsey. *Integer Programming*. John Wiley & Sons, 1998.
- S. Zoraster. The solution of large 0-1 integer programming problems encountered in automated cartography. *Operations Research*, 38:752–759, 1990.