

C 18847432

043  
384**UNIVERSIDAD DE SEVILLA**

Departamento de Matemática Aplicada I

UNIVERSIDAD DE SEVILLA  
FACULTAD DE MATEMÁTICAS  
BIBLIOTECA**INMERSIONES ORTOGONALES****DE GRAFOS****EN SUPERFICIES NO PLANAS**

Memoria presentada por  
María de los Ángeles Garrido Vizuet  
para optar al grado de  
Doctora en Matemáticas  
por la Universidad de Sevilla

Vº Bº  
del Director:

Fdo. Alberto Márquez Pérez  
Catedrático de Universidad  
del Departamento de  
Matemática Aplicada I  
de la Universidad de Sevilla

Sevilla, Julio de 1997

*A Ede y Paula*

*El Tajo es más bello que el río que pasa por mi aldea,  
pero el Tajo no es más bello que el río que pasa por mi aldea  
porque el Tajo no es el río que pasa por mi aldea.*

*En el Tajo hay grandes navíos,  
y todavía lo navega,  
para quienes en todo ven lo que no está,  
la memoria de las carabelas.*

*El Tajo baja de España  
y entra en la mar por Portugal.*

*Todos lo saben.*

*Mas poco saben cuál es el río de mi aldea  
y a donde va  
y de donde viene.*

*Y por eso, porque pertenece a menos gente,  
es más libre y mayor el río de mi aldea.*

*Por el Tajo se va al Mundo.*

*Más allá del Tajo está América  
y la fortuna de quienes la encuentran.*

*Nunca nadie ha pensado en lo que hay más allá  
del río de mi aldea.*

*El río de mi aldea no hace pensar en nada.  
Quien está junto a él sólo está junto a él.*

*Fernando Pessoa*

# Agradecimientos

A todos los que han aportado su granito, o montón de arena, en la realización de esta memoria, ofreciéndome conocimientos, tiempo y apoyo.

Gracias.

# Resumen

El área de investigación sobre dibujos de grafos (*graph drawing*) se ha convertido en un campo extensamente estudiado, constituyendo una interesante conexión entre la computación y la teoría de grafos. Dentro de ella, la representación ortogonal de grafos ocupa un lugar importante por su aplicación al diseño de circuitos VLSI, dando lugar a diversos problemas de optimización. Esta memoria está dedicada al estudio de las inmersiones ortogonales de grafos en superficies, con el objeto de minimizar el número de codos.

Motivados por el trabajo de Tamassia realizado en el plano, nos planteamos el problema de caracterizar la inmersión ortogonal cilíndrica que presenta el mínimo número de codos, entre todas las equivalentes a la de partida. Comenzamos realizando la caracterización de la asignación ortogonal óptima, concepto que recoge la información de los ángulos y codos del trazado, obteniéndola en tiempo polinomial.

En el plano, los conceptos de asignación ortogonal e inmersión en la malla son equivalentes. Sin embargo, en el cilindro la situación es muy distinta, por lo que realizamos un estudio detallado de la relación entre ambas estructuras, destacando las diferencias con respecto al plano. Igualmente, diseñamos algoritmos efectivos que proporcionan inmersiones ortogonales cilíndricas, de forma que el número total de codos obtenido constituye una buena aproximación del valor óptimo, guiados por dos enfoques, uno global en el grafo y otro local en cada arista.

Desde un punto de vista práctico, es necesario el estudio de inmersiones ortogonales de grafos en superficies. Este hecho nos conduce al planteamiento de dos importantes problemas en este ámbito, demostrando su naturaleza NP-completa: dada una inmersión en una superficie, decidir si admite una inmersión ortogonal sin codos esencialmente equivalente; y por otra parte, encontrar una superficie en la que un grafo dado admita una inmersión ortogonal sin codos. Dada la intratabilidad computacional de estos problemas, para obtener soluciones óptimas, presentamos algoritmos lineales de construcción de inmersiones ortogonales en superficies arbitrarias, que constituyen buenas aproximaciones a la solución óptima.

# Introducción

De un extenso campo de aplicaciones industriales ha surgido la necesidad del dibujo y visualización de grafos. Entre ellas podemos citar la planificación y supervisión de la producción, la elaboración de base de datos o el manejo de redes [72, 46, 75]. Por ejemplo, en esta última la red tiene que estar distribuida de forma que los puntos de riesgo potencial resulten fácilmente accesibles para los servicios de urgencia, como bomberos o ambulancias. De igual forma, cada situación presentará la necesidad de verificar diversas propiedades, que se traducen en condiciones sobre el tipo de trazado del grafo.

El Teorema de Kuratowski [44] constituye, en buena medida, al menos desde un punto de vista teórico, el inicio de esta disciplina. Data de 1930 y en él se caracterizan los grafos que admiten una inmersión en el plano. Este resultado no condujo a un algoritmo efectivo, teniendo que esperar varios años para conseguirlo. Para los grafos no planos, ya desde entonces se conocía la existencia de alguna superficie de mayor género, en la que el grafo se podía representar.

A finales de los años 60 empezó a surgir la necesidad de algoritmos para la representación de grafos, debido al incremento, cada vez mayor, del número de elementos que componían un circuito y que hacían el diseño demasiado complicado. En el libro de Lengauer [47] podemos encontrar una recopilación de estos primeros algoritmos para el diseño de redes. Fue entonces, cuando Hopcroft y Tarjan [37] (1974) obtienen el correspondiente resultado práctico al

Teorema de Kuratowski de planaridad, dando una caracterización que condujo al diseño de un algoritmo efectivo.

La evolución de este área derivó, a finales de los 80, en un gran interés por el diseño de algoritmos para obtener representaciones “estéticamente agradables” de grafos. Este hecho estuvo motivado por la ingeniería y los procesos de producción, en los que se llegó a la conclusión de que la información sobre el proceso constituía una importante fuente de datos para su control. Se necesitaban por tanto, herramientas que estructuraran la información, de forma que ésta fuera fácilmente comprensible. La representación de grafos constituía una muy buena herramienta para ello. Fruto de este interés, se obtienen los trabajos de Nishizeki y Chiba [17] (1985) y otros, para la representación de grafos planos en tiempo lineal.

En función de la aplicación, se han desarrollado diferentes formas de dibujar un grafo y se ha dedicado mucho esfuerzo en el diseño de buenos algoritmos de representación. Como muestra de ello podemos citar [5, 66, 6, 14, 56, 78]. Hoy en día, el campo de representación de grafos constituye una fructífera línea de investigación, con resultados importantes, tanto teóricos como computacionales.

Esta memoria se centra en un tipo concreto de representación, como es el trazado ortogonal, en el cual las aristas se deben dibujar a lo largo de líneas de una malla ortogonal, lo que da lugar a un ángulo mínimo entre aristas de 90 grados. Este hecho hace que el dibujo sea altamente manejable, aunque contenga un número elevado de aristas. Este tipo de trazado se utiliza en el diseño de circuitos, donde la ortogonalidad es un requisito tecnológico, y en aplicaciones de base de datos.

La definición general y precisa de *dibujo ortogonal* es la siguiente. Cada vértice está representado por una caja en el espacio  $\mathbf{R}^n$  con lados paralelos a los ejes coordenados y cada arista por una secuencia de segmentos consecutivos conectando las cajas correspondientes a sus extremos. Cada segmento tiene que ser paralelo a uno de los ejes coordenados. El punto donde una arista



- **Tamaño de los vértices:** Es importante mantener un control sobre el tamaño de los vértices si éstos se representan mediante cajas. Es interesante saber si es posible dibujar un grafo sin codos, aumentando el tamaño de los vértices; sin embargo cajas muy grandes producen efectos visuales negativos, por lo que se pretende mantener un tamaño pequeño para las cajas.

Estos cuatro elementos, junto con el espacio en el que consideremos la representación, conducen a diferentes problemas sobre dibujos ortogonales.

En general, en los tipos de representaciones ortogonales consideradas en la literatura cada vértice está asociado a un punto del espacio, aunque también podemos encontrar trabajos con cajas representando los vértices [24, 9]. Nuestra visión se realizará desde el primer punto de vista.

Una condición necesaria y suficiente para que un grafo plano admita una inmersión ortogonal plana es que sea un 4-grafo, es decir que las valencias de sus vértices sean menores o iguales a cuatro. En [61, 75] se prueba que todo 4-grafo plano admite una inmersión ortogonal plana con área  $O(n^2)$  y se presentan grafos que requieren área cuadrática. Storer [63] y Tamassia y Tollis [67] obtienen inmersiones ortogonales planas de grafos en una malla de tamaño  $n \times n$  con  $2n + 4$  codos si son 2-conexos, y  $2.4n + 2$  en otro caso. El número de codos a lo largo de cada arista es cuatro, como máximo.

Estos resultados proporcionan inmersiones ortogonales planas de grafos, en las que se ha computado el área y número de codos que presentan. Puesto que la situación óptima correspondería a los valores mínimos de estos elementos, así como del número de cruces si no se trata de un grafo plano, surgen los correspondientes problemas de optimización, obteniendo la naturaleza intratable de ellos. El problema de determinar si un grafo admite una inmersión rectilínea es NP-completo [31] y lo mismo ocurre para decidir si un grafo se puede representar sobre una malla de tamaño fijado [27, 43]. Los resultados sobre la NP-dureza de la minimización del área y de la longitud

---

total de las aristas en representaciones ortogonales de grafos quedan recogidos en [63, 8, 13, 32, 38, 21]. Igualmente, el problema de minimizar el número de cruces es también NP-duro [30]. Estos resultados dirigen las investigaciones hacia heurísticas que sean razonablemente efectivas, introduciendo restricciones en los problemas. Di Battista, Liotta y Vargiu [7] demuestran que para la familia de grafos 3-planos se puede determinar en tiempo polinomial una inmersión ortogonal plana con el mínimo número de codos.

Prácticamente, la mayoría de los trabajos realizados hasta ahora se han centrado en representaciones de grafos en  $\mathbf{R}^2$ , bien sea para grafos planos o no. Podemos encontrar algunos resultados sobre representación de grafos en  $\mathbf{R}^3$  [58]. Sin embargo, si retomamos el origen de esta línea de investigación, nos encontramos con que el diseño de circuitos, desde un punto de vista práctico, lleva a considerar grafos no planos, los cuales admiten una inmersión en otro tipo de superficie, determinando la necesidad de evitar cruces inapropiados, para lo que hay que realizar orificios de una cara a otra de la placa. También se requieren conexiones entre nodos situados en los bordes. Todo esto hace que la modelización plana de una placa sea inapropiada. Este hecho, junto con las *representaciones ortogonales estándares* de las superficies, introducidas por Cobos [19], nos conducen al trazado de grafos en superficies distintas al plano. Este tipo de representación de las superficies proporciona dos familias de rectas ortogonales, por lo que el concepto de dibujo ortogonal es fácilmente generalizable. Estos incentivos nos llevan al planteamiento de problemas sobre inmersiones ortogonales en superficies, quedando recogida su naturaleza, también NP-dura, en esta memoria.

La intratabilidad computacional de los problemas anteriormente citados, además de conducir a diversos resultados de aproximación, ha originado una variante que, en lugar de plantear aspectos ortogonales de un grafo dado, considera una inmersión plana de un 4-grafo y se pregunta por una inmersión ortogonal equivalente que presente el mínimo número de codos. Para este problema, Tamassia [64] ha obtenido un algoritmo  $O(n^2 \log n)$  que lo resuelve. En esta memoria se realiza el mismo planteamiento en el cilindro, obteniéndose

resultados de caracterización teóricos y algoritmos efectivos.

En cuanto al número de codos, dos líneas determinaban su tratamiento, una global en el grafo y otra local en cada arista. El algoritmo  $O(n^2 \log n)$  de Tamassia se rige por la primera y su implementación, realizada en la Universidad de Roma “La Sapienza”, ha dado lugar a inmersiones óptimas pero con un número elevado de codos en diversas aristas. Podemos encontrar otras heurísticas [40, 50, 51, 12, 53, 10] guiadas por la línea local, en las que se pretende que el número de codos en cada arista sea lo menor posible y en las que se reduce el tiempo de computación a lineal. En algunas se tratan con grafos 3-planos y 4-planos 3-conexos, obteniéndose en algunos casos inmersiones ortogonales con, únicamente un codo en cada arista. En [11] se conserva la inmersión dada al pasar a la malla ortogonal plana.

Hemos diseñado algoritmos polinomiales para inmersiones ortogonales en el cilindro y en superficies en general, realizando el cómputo del número de codos obtenido, que para el caso cilíndrico constituye una mejora con respecto al plano.

La presente memoria se ha dividido en dos partes, en función de la superficie sobre la que se trazan las inmersiones ortogonales. Recogiendo los conceptos básicos necesarios para ambas partes hemos incluido un capítulo de Preliminares, con el cual el lector puede situarse en el contexto en el que se desarrolla el trabajo.

La primera parte de la memoria, inmersiones ortogonales cilíndricas, está compuesta por dos capítulos, siendo el cilindro la superficie base de ambos. En el primer capítulo se parte de una inmersión cilíndrica no plana, y se realiza un estudio exhaustivo de sus asignaciones ortogonales asociadas, conjunto de ángulos y codos, proporcionando un algoritmo polinomial para construir la asignación asociada que presenta el mínimo número de codos. El planteamiento de este problema fue motivado por una visita del profesor Tamassia, como generalización de su trabajo presentado en [64].

En el segundo y último capítulo de esta parte se trata la conversión de una asignación ortogonal a una inmersión ortogonal cilíndrica que conserve la inmersión de partida. Este paso en el plano es automático, sin embargo demostramos que en el cilindro es totalmente diferente, dando lugar a asignaciones ortogonales realizables y no realizables. Presentamos su caracterización y dentro de las realizables determinamos, de forma teórica, la que proporciona el mínimo número de codos. Para la generación de inmersiones ortogonales cilíndricas, presentamos un algoritmo, que de forma global en el grafo, y en tiempo polinomial, aproxima el número de codos conservando la inmersión de partida. Para reducir el tiempo de computación y bajo una visión local en cada arista, demostramos que todo 4-grafo admite una inmersión ortogonal en el cilindro con, a lo más, dos codos en cada arista, siendo mínimo este valor dos, mejorándose así los resultados conocidos para el plano. Para grafos 2-conexos, un algoritmo lineal proporciona una tal inmersión, partiendo de una cilíndrica y siendo éstas equivalentes.

En la segunda parte de la memoria el planteamiento es más general. Trabajamos con superficies de género arbitrario, utilizando las representaciones ortogonales estándares de éstas, introducidas en el capítulo de preliminares. Estas representaciones de las superficies proporcionan relaciones de ortogonalidad entre dos familias de rectas, haciendo posible el tratamiento de inmersiones ortogonales. Planteamos dos problemas de minimización del número de codos demostrando su naturaleza NP-dura.

En el primer capítulo de esta parte introducimos el concepto de inmersiones esencialmente equivalentes, para distinguir entre subgrafos esenciales y no esenciales en una inmersión. Probamos la naturaleza NP-completa del problema de decidir si, dada una inmersión en una superficie, admite una rectilínea esencialmente equivalente.

Garg y Tamassia [31] plantearon el problema de determinar si un grafo plano admite una inmersión rectilínea plana, demostrando su naturaleza NP-completa. Si no nos restringimos al plano, sino que trabajamos con superficies

generales, se obtiene el problema tratado en el último capítulo de la memoria. Partimos de un grafo, con el fin de determinar una inmersión ortogonal de él que minimize el número de codos entre todas las posibles en cualquier superficie. Nos encontramos con la intratabilidad del problema, al probar su naturaleza NP-dura, la cual nos conduce al diseño de algoritmos eficientes para construir inmersiones ortogonales en superficies con un número total de codos que constituye una aproximación local del valor óptimo. Las complejidades de estos algoritmos dependen linealmente del número de vértices del grafo y del género de la superficie.

# Contenido

<b>Agradecimientos</b>	<b>i</b>
<b>Resumen</b>	<b>iii</b>
<b>Introducción</b>	<b>v</b>
<b>Contenido</b>	<b>xiii</b>
<b>0 Preliminares</b>	<b>1</b>
0.1 Introducción . . . . .	1
0.2 Notaciones en Teoría de Grafos . . . . .	2
0.3 Inmersiones de grafos en superficies . . . . .	7
0.4 Aspectos algorítmicos . . . . .	20
0.5 NP-completitud . . . . .	22

---

<b>I</b>	<b>Inmersiones ortogonales cilíndricas</b>	<b>31</b>
<b>1</b>	<b>Asignaciones ortogonales cilíndricas</b>	<b>35</b>
1.1	Introducción . . . . .	36
1.2	Asignaciones ortogonales de inmersiones cilíndricas . . . . .	44
1.2.1	Aspectos algorítmicos . . . . .	67
1.3	Conclusiones y problemas abiertos . . . . .	71
<b>2</b>	<b>Inmersiones ortogonales cilíndricas</b>	<b>73</b>
2.1	Introducción . . . . .	74
2.2	Asignaciones ortogonales realizables . . . . .	76
2.3	Aproximación en el número de codos . . . . .	113
2.3.1	Globalmente . . . . .	114
2.3.2	Localmente . . . . .	123
2.4	Conclusiones y problemas abiertos . . . . .	132
<b>II</b>	<b>Inmersiones ortogonales en superficies</b>	<b>135</b>
<b>3</b>	<b>Inmersiones ortogonales esencialmente equivalentes</b>	<b>139</b>
3.1	Introducción . . . . .	139
3.2	Inmersiones rectilíneas esencialmente equivalentes . . . . .	141
3.3	Conclusiones y problemas abiertos . . . . .	154

---

<b>4 Inmersiones ortogonales de grafos en superficies</b>	<b>157</b>
4.1 Introducción . . . . .	158
4.2 Minimizar el número de codos es NP-duro . . . . .	159
4.3 Inmersiones ortogonales con pocos codos . . . . .	166
4.4 Conclusiones y problemas abiertos . . . . .	172
 <b>Referencias</b>	 <b>175</b>
 <b>Índice</b>	 <b>182</b>

# Capítulo 0

## Preliminares

### 0.1 Introducción

El presente capítulo se dedicará a dar las definiciones generales y establecer algunas de las notaciones que se utilizarán a lo largo de esta memoria. La materia que tratamos se puede enmarcar bajo dos contextos generales muy relacionados, como son la Teoría de Grafos y la Algorítmica. Por una parte tratamos nociones básicas de grafos, desde su definición y propiedades, hasta el concepto de inmersión plana, para después generalizarlo a superficies de género mayor, proporcionando una nueva representación de éstas. Por otro lado reflejamos aspectos algorítmicos e introducimos la Teoría de la NP-completitud.

Sería recomendable para una completa comprensión de la memoria, sobre todo para aquellos lectores no familiarizados con la materia, la lectura de textos básicos como pueden ser el de Harary [35] de Teoría de Grafos, Aho, Hopcroft y Ullman [1] de Algorítmica, Garey y Johnson [29] sobre NP-completitud o los de Nishizeki y Chiba [55] y Chartrand y Lesniak [16] de Algorítmica aplicada a la Teoría de Grafos. Los textos citados hasta ahora se pueden considerar como clásicos dentro de su materia. En cuanto a inmersiones

de grafos en superficies de mayor género podemos nombrar los de White [76], Gross y Tucker [33] y como más reciente el debido a Liu Yanpei [78].

## 0.2 Notaciones en Teoría de Grafos

Como hemos indicado, comenzamos introduciendo notaciones en Teoría de Grafos.

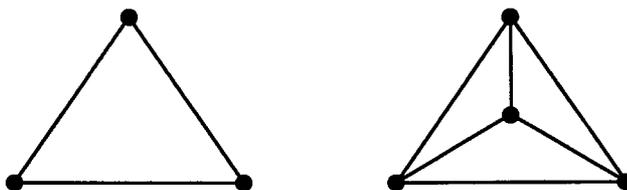
Se define un *grafo*  $G$  como un par  $(V, A)$  formado por un conjunto de *vértices*  $V$  y una colección de *aristas*  $A$ , donde cada arista  $l \in A$  es un par no ordenado de vértices distintos de  $G$ , es decir,  $l = \{v_i, v_j\}$  con  $v_i, v_j \in V$ . Dos aristas son *incidentes* si comparten un vértice. Un vértice  $v$  y una arista son *incidentes* si el vértice  $v$  es uno de los que definen a dicha arista. Dos vértices se dicen *adyacentes* si ambos definen una arista de  $G$ . Se denota por  $n$  y  $m$  al número de vértices y aristas de un grafo, respectivamente.

Si en el grafo las aristas están formadas por pares ordenados de vértices obtenemos un *grafo dirigido* o *digrafo*.

Como caso más general, si un vértice es adyacente a él mismo decimos que en  $A$  existe un *lazo*. Si dos vértices son adyacentes a través de más de una arista decimos que existen en  $A$  *aristas múltiples*. Llamamos *multigrafo* a un grafo en el que se permiten lazos y aristas múltiples.

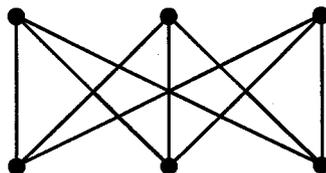
Dado un grafo  $G = (V, A)$  se denomina *valencia* del vértice  $v \in V$  al número de aristas de  $G$  que inciden en dicho vértice, y se denota por  $\delta(v)$ . Si el grafo  $G = (V, A)$  posee  $n$  vértices y todos ellos poseen la misma valencia, se dice que es *regular*. Como caso particular, si la valencia de cada vértice es  $n - 1$ , el grafo no admite más aristas y se dice que se trata de un *grafo completo*, denotándose por  $K_n$ , donde  $n$  representa el número de vértices (ver Figura 0.1).

Llamaremos *k-grafo* a aquél cuyos vértices tengan valencia a lo más  $k$ .

Figura 0.1: Los grafos completos  $K_3$  y  $K_4$ .

En el caso en el que todos los vértices tengan la misma valencia e igual a  $k$  diremos que el grafo es  $k$ -valente.

Un grafo  $G=(V, A)$  se dice que es *bipartito* si podemos hacer una partición del conjunto  $V$  en dos subconjuntos  $V_1$  y  $V_2$  de tal forma que cualquier arista  $l \in A$  está definida por un vértice de  $V_1$  y otro de  $V_2$ . Un grafo bipartito de  $m+n$  vértices ( $m$  vértices en  $V_1$  y  $n$  en  $V_2$ ) tal que al añadirle una nueva arista deja de ser bipartito, se dice que es un grafo *bipartito completo* y se denota por  $K_{m,n}$ .

Figura 0.2: El grafo bipartito completo  $K_{3,3}$ .

Dados dos vértices  $v_1$  y  $v_2$  de un grafo  $G$ , un *camino* en  $G$  que une  $v_1$  con  $v_2$  es un conjunto ordenado de vértices  $\alpha = \{v_1, w_1, \dots, w_k, v_2\}$  con la condición de que cada vértice sea adyacente al anterior y al posterior. Definiremos la longitud de un camino  $\alpha$  como el número de vértices que posee menos uno. También se llama camino en  $G$  de  $v_1$  a  $v_2$  a la sucesión de aristas  $\{\{v_1, w_1\}, \{w_1, w_2\}, \dots, \{w_k, v_2\}\}$ . La longitud de este camino (que es la misma que la de  $\alpha$ ) viene determinada por el número de aristas que lo componen.

Un camino se dice que es *simple* si no contiene vértices repetidos. Resulta fácil comprobar que si existe un camino que une a los vértices  $v_i$  y  $v_j$ , existe un camino simple que también los une y por ello, siempre que hablemos de un camino nos estaremos refiriendo a uno simple. La Figura 0.3 muestra un camino de longitud 4.

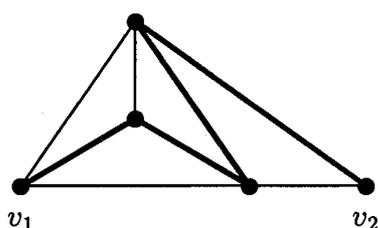


Figura 0.3: Un camino de longitud 4 entre los vértices  $v_1$  y  $v_2$ .

Un *ciclo* es un camino que comienza y termina en el mismo vértice, diciéndose que es *simple* si los únicos vértices que se repiten son el primero y el último. En general al referirnos a un ciclo lo haremos a uno simple. Al grafo compuesto por un único ciclo de longitud  $n$  se le llama *grafo cíclico* y se le denota  $C_n$  (ver Figura 0.4).



Figura 0.4: Los grafos cíclicos  $C_3$  y  $C_4$ .

Notemos que los grafos  $K_3$  y  $C_3$  coinciden.

Sea  $\mathcal{R}$  la relación de equivalencia entre los vértices de un grafo  $G$ ,

definida por  $v \mathcal{R} w$  si y sólo si existe un camino en  $G$  entre  $v$  y  $w$ . Llamaremos *componentes conexas* (o simplemente componentes) de  $G$  a las clases de equivalencia del conjunto cociente  $G/\mathcal{R}$ , y diremos que un grafo es *conexo* si sólo tiene una componente conexa.

A lo largo de la memoria consideraremos grafos conexos, ya que de lo contrario razonaríamos con cada una de sus componentes conexas.

Se dice que el grafo  $G' = (V', A')$  es un *subgrafo* de  $G = (V, A)$  si  $V' \subseteq V$  y  $A' \subseteq A$ . Dado un grafo  $G = (V, A)$  y un conjunto  $W$  de vértices de  $G$ , se define el *subgrafo inducido* por  $G - W$  como el subgrafo de  $G$  cuyo conjunto de vértices es  $V' = V - W$  y cuyas aristas son las aristas de  $G$  definidas por parejas de vértices de  $V'$ . Diremos también que el grafo inducido por  $G - W$  es el grafo resultante de *eliminar* de  $G$  los vértices de  $W$ .

Un vértice  $v$  de un grafo  $G$  se dice que es un *punto de corte* si el grafo inducido por  $G - \{v\}$  posee un número superior de componentes conexas que el grafo  $G$ . Un grafo conexo que no posee puntos de corte se dice que es un grafo *2-conexo*.

Si  $G$  es un grafo conexo y  $G'$  es un subgrafo 2-conexo de  $G$ , se dice que  $G'$  es *maximal* si cualquier subgrafo de  $G$  que contenga a su vez a  $G'$  como subgrafo, contiene puntos de corte. A los subgrafos maximales 2-conexos de un grafo  $G$  se les denomina *bloques* de  $G$ . Obsérvese que cada bloque  $B$  de un grafo  $G$  contiene a todos los puntos de corte adyacentes a vértices de  $B$ .

Si  $G$  es un grafo 2-conexo, evidentemente, no tiene puntos de corte y, por tanto, el grafo resultante de *eliminar* cualquier vértice de  $G$  resulta ser conexo. Sin embargo, puede ocurrir que al eliminar *dos* de sus vértices se rompa la conexión. En ese caso, diremos que dicho par de vértices constituyen una *pareja de corte*. Un grafo 2-conexo que no posea parejas de corte se dice que es *3-conexo*.

En general, para romper la conexión de un grafo conexo será necesario

eliminar un conjunto de, al menos,  $k$  vértices. A este conjunto de  $k$  vértices se le denomina *conjunto de corte* y diremos que el grafo es  $k$ -conexo. Podemos observar ejemplos de grafos con distintos grados de conexión en las Figuras 0.5 y 0.6.

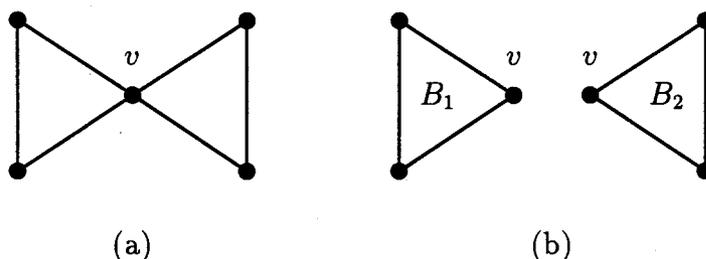


Figura 0.5: (a) Un grafo conexo con un punto de corte  $v$ , (b) y sus bloques.

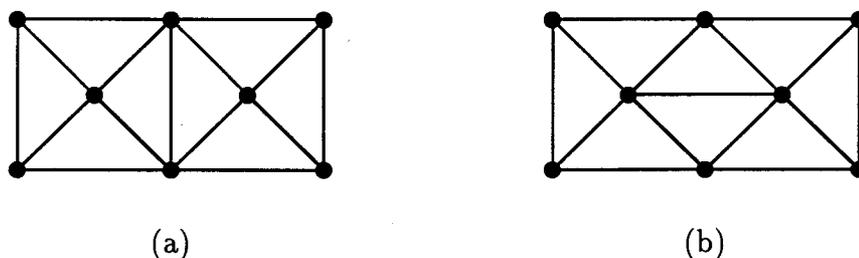


Figura 0.6: (a) Un grafo 2-conexo no 3-conexo, (b) y otro 3-conexo.

Si  $G_1 = (V_1, A_1)$  y  $G_2 = (V_2, A_2)$  son dos grafos y  $\phi : V_1 \rightarrow V_2$  es una aplicación inyectiva de tal forma que si los vértices  $v$  y  $w$  de  $G_1$  son adyacentes en  $G_1$  entonces  $\phi(v)$  y  $\phi(w)$  son adyacentes en  $G_2$ , se dice que  $\phi$  es un *homomorfismo* de  $G_1$  en  $G_2$ . Si  $\phi$  es un homomorfismo biyectivo de  $G_1$  en  $G_2$  de forma que  $\phi^{-1}$  es un homomorfismo de  $G_2$  en  $G_1$  entonces se dice que  $\phi$  es un *isomorfismo* entre  $G_1$  y  $G_2$ . En tal caso, se dice que  $G_1$  y  $G_2$  son *isomorfos*.

Las definiciones dadas anteriormente para grafos se pueden generalizar a digrafos y multigrafos.

Dado un grafo  $G$  decimos que la numeración  $1, 2, \dots, n$  de sus vértices es una *st-numeración* si los vértices  $1$  y  $n$  son adyacentes y para cada  $j$ ,  $2 \leq j \leq n - 1$ , existen dos vértices  $i$  y  $k$  adyacentes con  $j$  tal que  $i < j < k$ . Al vértice  $1$  se le llama *fuentes* y se le denota  $s$  mientras que al  $n$  se le llama *sumidero* denotándolo por  $t$ . La Figura 0.7 muestra una *st-numeración* del grafo dado. Even y Tarjan [26] demuestran que todo grafo 2-conexo admite una *st-numeración* y presentan un algoritmo lineal en el número de vértices y aristas del grafo para encontrarla.

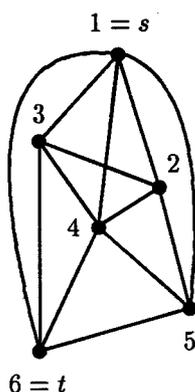


Figura 0.7: Una *st-numeración* de un grafo.

### 0.3 Inmersiones de grafos en superficies

El diseño de circuitos impresos es un claro incentivo para la representación de grafos en distintas superficies. Muchas cuestiones se han planteado sobre dibujos planos de grafos que han derivado en diversos e importantes resultados. No han ocupado un lugar menos significativo las investigaciones realizadas sobre otras superficies dando lugar a una amplia literatura sobre el tema.

Nuestro objetivo en esta sección consiste en fijar los conceptos sobre superficies e inmersiones de grafos, con los que trabajaremos a lo largo de

la memoria, e introducir una nueva representación de las superficies que nos permitirá abordar nuestros problemas.

Suponemos conocidos conceptos básicos como espacio topológico, compacidad, conexión por arcos, espacio de Hausdorff, homeomorfismo, etc. En este contexto decimos que una *superficie*  $S$  es un espacio topológico compacto, de Hausdorff, conexo por arcos y localmente homeomorfo a un disco, esto es, para cada punto  $p$  de  $S$  existe un conjunto abierto de  $S$  conteniendo a  $p$ , el cual es homeomorfo a un disco abierto del plano. Una superficie se puede obtener de la siguiente forma: tomamos una colección de polígonos convexos en el plano disjuntos dos a dos (y sus interiores) con lados de longitud uno e identificamos cada lado con otro, posiblemente en el mismo polígono. Esto da lugar a un espacio topológico  $S$  y a un grafo  $G$ , cuyos vértices son los de los polígonos y cuyas aristas son los lados. Si  $G$  es conexo,  $S$  es conexo por arcos y si  $S$  es localmente homeomorfo a un disco en cada uno de los vértices de  $G$ , entonces  $S$  es una superficie. Decimos que  $G$  es una *inmersión 2-celular* de  $S$ .

Definimos seguidamente las superficies  $S_g$ . Denotamos por  $S_0$  a la esfera. Si recortamos dos discos abiertos disjuntos de  $S_0$  e identificamos sus fronteras de tal forma que las orientaciones en el sentido de las agujas del reloj de éstas no coincidan, estamos añadiendo un *asa* a  $S_0$ . Si añadimos  $g$  asas obtenemos la *superficie orientable*  $S_g$ .  $S_1$  representa la superficie tórica y  $S_2$  el doble toro. Al número de asas de una superficie  $S$  se le llama *género* de  $S$ .

Como caso particular de la construcción anterior, la superficie orientable de género  $g$ , o *superficie poligonal estándar*, la podemos construir identificando los lados de un  $4g$ -polígono como nos indica la Figura 0.8, dando lugar a sus  $g$  asas.

Enunciamos así, el conocido Teorema de Clasificación de Superficies:

**Teorema 0.3.1.** [28] *Toda superficie orientable es homeomorfa a una de las superficies poligonales estándares.*

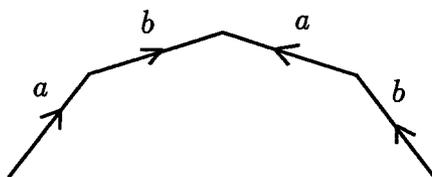


Figura 0.8: Una de las asas de una superficie de género  $g$ .

En este punto, podemos considerar representaciones topológicas de grafos en distintas superficies. Un *dibujo* de un grafo  $G$  en una superficie  $S$  es aquél en el que los vértices están representados por puntos y las aristas por curvas de  $S$ . Decimos que  $G$  está *inmerso* o admite una *inmersión* en  $S$  si es posible obtener un dibujo de tal forma que dos aristas cualesquiera no se intersecten en puntos que no sean vértices comunes a ambas. Si consideramos el complementario en  $S$  de una inmersión de un grafo obtenemos una serie de regiones distintas llamadas *caras* de la inmersión. El conjunto de aristas que delimitan una cara se le llama *frontera de la cara*.

Dada una inmersión  $P$  de un grafo en una superficie, su *dual geométrico*  $P^*$  se define como sigue: se sitúa un vértice en cada cara de  $P$  y por cada arista  $x$  compartida por dos caras se define en  $P^*$  una arista entre los vértices asociados a dichas caras, cruzando únicamente la arista  $x$ .

La relevancia del plano como caso particular dentro de las superficies hace que demos las siguientes definiciones.

Un grafo  $G = (V, A)$  se dice que es *plano* si admite una inmersión plana. Considerando la proyección estereográfica representada en la Figura 0.9, es fácil ver que un grafo es plano si y sólo si admite una inmersión en la esfera, es decir también es un *grafo esférico*.

Diremos que un grafo es *4-plano* si es un 4-grafo y plano.

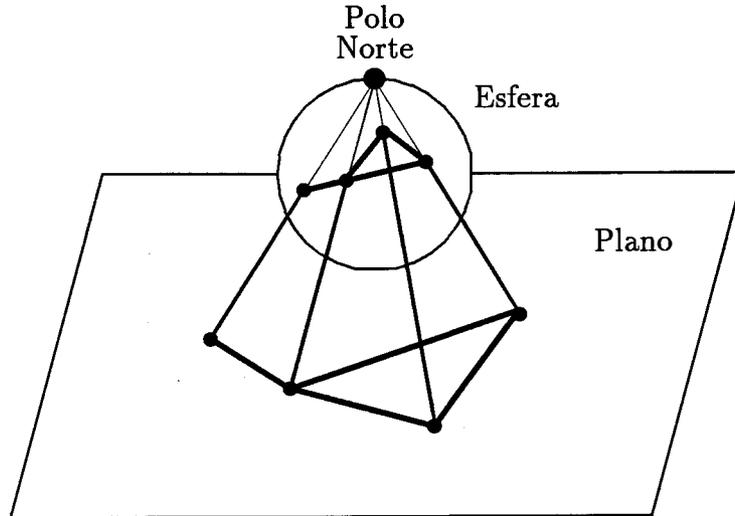


Figura 0.9: Proyección estereográfica.

Nos encaminamos seguidamente a enunciar uno de los teoremas más importantes en Teoría de Grafos, el Teorema de Kuratowski, que caracteriza los grafos planos.

Si  $l = \{v_1, v_2\}$  es una arista de un grafo  $G = (V, A)$ , llamaremos *partición* de la arista  $l$  al resultado de sustituir dicha arista por el camino simple  $\{v_1, w_1, \dots, w_k, v_2\}$  donde los vértices  $w_i$ , para  $1 \leq i \leq k$ , son todos distintos y ninguno de ellos perteneciente a  $V$ .

Dado un grafo  $G$  se define una *subdivisión* de  $G$  al grafo resultante de realizar sucesivas particiones de algunas de sus aristas. Diremos que dos grafos  $G_1$  y  $G_2$  son *homeomorfos* si existe un tercer grafo  $G_3$  que sea subdivisión de  $G_1$  y  $G_2$ .

**Teorema 0.3.2.** [44] *Un grafo  $G$  es plano si y sólo si no contiene a ningún subgrafo homeomorfo a  $K_5$  ni a  $K_{3,3}$ .*

Volviendo al estudio general de inmersiones de grafos en superficies, nos

centramos en la relación existente entre el número de vértices, aristas y caras de una inmersión. Aunque de todos es conocida como la fórmula de Euler, son varios los autores que hay que citar al hablar de dicha fórmula. En 1750 Euler escribió una carta a Goldbach comunicándole que había encontrado una nueva fórmula relacionando el número de vértices  $n$ , aristas  $m$  y caras  $f$  de un poliedro (esférico), pero que no había sido capaz de probarla.

$$n - m + f = 2$$

La carta completa se puede encontrar en [39]. Cauchy [15] en 1813 dio una prueba, básicamente teórica de la fórmula pero fue en 1811 cuando Lhuillier [48] clasificó las aparentes excepciones y generalizó la fórmula de Euler a superficies cerradas orientables, resultado que recogemos en el siguiente teorema:

**Teorema 0.3.3.** [48] *Si  $G$  es una inmersión 2-celular de una superficie orientable  $S_g$ , se tiene que*

$$n - m + f = 2 - 2g$$

donde  $n$ ,  $m$  y  $f$  denotan, respectivamente, el número de vértices, aristas y caras de  $G$ .

El número  $n - m + f$  no depende de  $G$  y se le llama *característica de Euler* de  $S$ .

De la fórmula de Euler se obtiene el siguiente resultado:

**Teorema 0.3.4.** [74] *Si  $G$  es un grafo con  $n$  vértices y  $m$  aristas inmerso en una superficie de género  $g$  entonces*

$$m \leq 3n - 6 + 6g.$$

La herramienta combinatoria básica para inmersiones de grafos en superficies orientables es una simple, pero importante y poderosa observación

debida a Heffter [36] (1891) y Edmonds [23] (1960). Sea  $G$  un grafo inmerso en  $S_g$ , el *sistema de rotación* en los vértices del grafo se define de la siguiente forma: sea  $p$  un vértice de  $G$ , la inmersión origina una permutación cíclica  $\pi_p$  de las aristas incidentes en  $p$  donde  $\pi_p(a)$  es el sucesor de  $a$  siguiendo el sentido de las agujas del reloj alrededor de  $p$ . A dicha permutación cíclica se le llama *rotación en  $p$* . Se puede observar un ejemplo en la Figura 0.10.

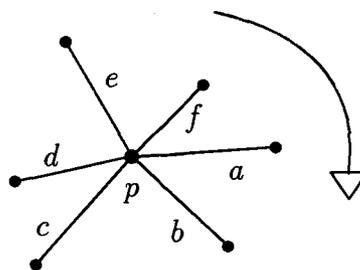


Figura 0.10: La rotación en  $p$  es la permutación cíclica  $abcdef$ .

Dos inmersiones de un grafo  $G$  en una superficie se dice que son *equivalentes* si sus sistemas de rotación coinciden.

Las superficies poligonales estándares constituyen una forma de representación de cualquier superficie orientable  $S_g$ , sin embargo la carencia de un sistema de referencia ortogonal en ellas, fundamental a lo largo de esta memoria para definir malla en cualquier superficie, hace que consideremos otra forma de representación equivalente a la dada.

Como indicamos anteriormente  $S_1$  representa la superficie tórica, la cual se puede obtener como figura geométrica girando una circunferencia  $Q$  de radio  $r$  alrededor de un eje situado a una distancia  $d > r$  del centro de  $Q$ . El conjunto  $C_1$  de las circunferencias que generan cada uno de los puntos de  $Q$  en su recorrido alrededor del eje, denominadas *paralelos* y el conjunto  $C_2$  de las circunferencias originadas por  $Q$  en cada momento de su giro alrededor del eje,

denominadas *meridianos*, constituyen un sistema de coordenadas ortogonal, ya que cada punto de la superficie del toro viene determinado unívocamente por la intersección de un paralelo con un meridiano (ver Figura 0.11).

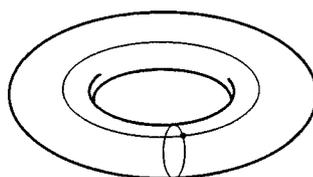


Figura 0.11: Sistema de coordenadas en el toro.

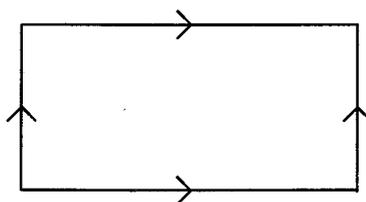


Figura 0.12: Toro plano.

Si *cortamos y desarrollamos* el toro por un paralelo y un meridiano obtenemos una representación como la indicada en la Figura 0.12, en la que los bordes superior e inferior así como el derecho y el izquierdo del rectángulo obtenido se encuentran respectivamente identificados. En virtud del homeomorfismo existente entre los puntos del toro y los del rectángulo, esta es una representación del toro sobre el plano que recibe el nombre de *toro plano*.

A partir del toro plano, superficie de género 1, Cobos [19] obtiene las de mayor género de la siguiente forma. Si *recortamos* en el toro plano una *ventana*, rectángulo de lados paralelos a los del toro plano, en el que identificamos el borde superior con el inferior así como el derecho con el izquierdo, obtenemos una superficie plana homeomorfa al doble toro (ver Figura 0.13) pues basta

con observar que la ventana que hemos introducido le agrega un asa al toro (una circunferencia que rodee a dicha ventana representará a la circunferencia de unión de las dos asas del doble toro).

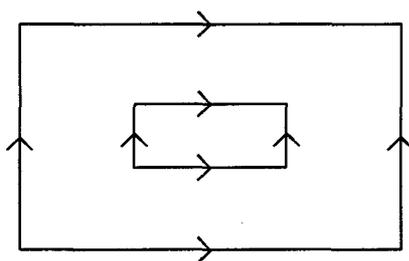


Figura 0.13: El doble toro.

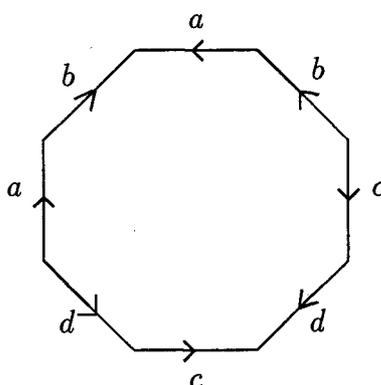


Figura 0.14: Representación poligonal del doble toro.

Podemos observar que los bordes del toro plano inicial y los de la ventana que hemos añadido se corresponden con un paralelo y un meridiano en cada una de las asas, como indica la Figura 0.15, al igual que los lados  $a$ ,  $b$ ,  $c$  y  $d$  del octógono  $S_2$ , superficie poligonal estándar asociada al doble toro (ver Figura 0.14), se corresponden con las curvas cerradas indicadas en la Figura 0.16.

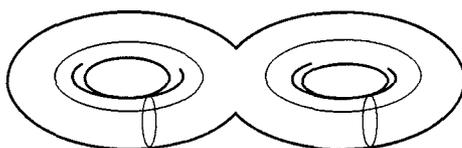


Figura 0.15: Las secciones en el doble toro que producen  $O_2$ .

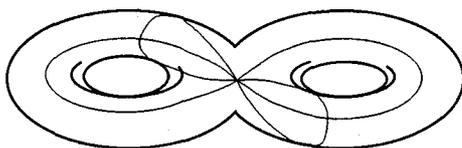


Figura 0.16: Las secciones en el doble toro que producen  $S_2$ .

En general, si denominamos *superficie ortogonal estándar* y denotamos  $O_g$  ( $g > 0$ ) a la superficie resultante de abrir  $g - 1$  ventanas en el toro plano (véase la Figura 0.17) e identificamos  $O_0$  con  $S_0$ , se puede dar el siguiente teorema general de clasificación de superficies:

**Teorema 0.3.5.** [19] *Toda superficie orientable y conexa es homeomorfa a una de las superficies ortogonales estándares  $O_0, O_1, O_2 \dots O_g$ .*

Podemos hablar de *rectángulos* en la superficie ortogonal estándar como los que delimitan las ventanas de la superficie junto con el rectángulo exterior.

Igual que en el toro plano, en toda superficie ortogonal estándar podemos hablar de dos familias de rectas perpendiculares entre sí, los paralelos y los meridianos. Los primeros se corresponden con la dirección horizontal y los segundos con la vertical. Estas dos familias de rectas determinan un sistema de

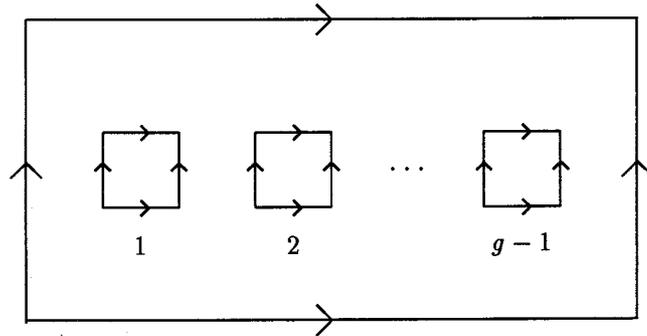


Figura 0.17: La superficie ortogonal estándar de género  $g$ .

referencia ortogonal en  $O_g$  y el conjunto de ambos tipos de rectas constituyen lo que llamaremos *mallá en la superficie*.

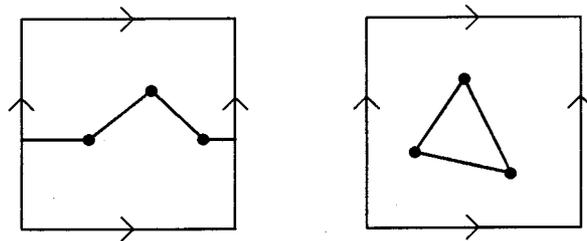


Figura 0.18: Inmersiones iguales combinatoriamente y distintas desde un punto de vista topológico.

Claramente las nociones de inmersión topológica e inmersión combinatoria no coinciden, tal y como muestra la Figura 0.18. Aprovechando la descripción de las superficies mediante las representaciones ortogonales estándares, podemos añadir alguna información suplementaria a la rotación en cada vértice que nos da la inmersión combinatoria, para obtener así la in-

formación necesaria para una descripción de la inmersión topológica. Una inmersión topológica de un grafo  $G$  en una superficie  $O_g$  determina un sistema de rotación en los vértices y etiqueta las aristas cuyas curvas asociadas en la superficie tienen intersección no vacía con algún lado de alguno de los rectángulos de  $O_g$ . Así, las herramientas que nos proporciona una inmersión en una superficie ortogonal estándar tienen doble naturaleza: rotación en los vértices y aristas concretas, llamémoslas *aristas esenciales*.

Diremos que un ciclo de la inmersión es *esencial* u *homotópicamente no nulo* si la curva cerrada asociada a dicho ciclo no encierra un disco abierto. Notemos que los ciclos esenciales están estrechamente relacionados con las aristas esenciales.

Dos inmersiones de un grafo  $G$  en una superficie son *equivalentes* si coinciden sus sistemas de rotación y sus ciclos presentan la misma naturaleza, esencial o no esencial.

Pasamos a dar los conceptos básicos, sobre los que trabajaremos en esta memoria. Una *inmersión ortogonal* de un grafo en la superficie ortogonal estándar de género  $g$  asocia a cada vértice un punto de la superficie y transforma cada arista en una cadena de segmentos consecutivos contenidos en paralelos o meridianos, de tal forma que dos cadenas asociadas a aristas distintas no se cruzan en puntos que no sean vértices. En definitiva, una inmersión ortogonal es una inmersión en la malla de  $O_g$ . Cada arista estará representada por una serie de segmentos consecutivos, horizontales y verticales, de forma que cada vez que se produzca un cambio de dirección en dicha cadena se origina un *codo* en la inmersión.

Una *inmersión rectilínea* de un grafo en  $O_g$  es una inmersión ortogonal en la que cada arista está representada por un único segmento horizontal o vertical, es decir, una inmersión ortogonal sin codos.

La forma más frecuente de representar un grafo, como acabamos de ver, es mediante una estructura de tipo topológico; los vértices se asocian a puntos

de la superficie y las aristas a arcos de curvas de Jordan con determinadas propiedades. Sin embargo, en diversas situaciones es recomendable utilizar otro tipo de representación que establezca las relaciones existentes entre sus vértices. La más frecuente, y cada vez considerada por más autores, es la relación de visibilidad obteniendo en dicho caso una *representación de visibilidad*. En una representación de este tipo los vértices vienen determinados por objetos que pueden, o no, impedir la visibilidad entre otros y las aristas vienen determinadas por ciertas relaciones de visibilidad entre sus vértices.

Entre las representaciones de visibilidad, la más conocida es la bar-visibilidad. En un *grafo de bar-visibilidad*, los vértices son segmentos verticales y dos vértices son adyacentes si las dos barras verticales que representan pueden verse la una a la otra horizontalmente y de forma no degenerada. En otras palabras, debe existir un rectángulo de altura no nula y limitado a derecha e izquierda por ambas barras de tal forma que no intersecte a ninguna de las demás barras verticales. Esta noción de visibilidad también es denominada, con frecuencia, como  $\varepsilon$ -visibilidad (Tamassia y Tollis [68]), ya que las barras deben ser capaces de verse entre ellas bajo un *rayo de visibilidad* de grosor  $\varepsilon > 0$ . La relación entre las representaciones de bar-visibilidad y las inmersiones ortogonales es muy fuerte puesto que las direcciones de los segmentos que constituyen los vértices (vertical) y de las relaciones de visibilidad (horizontal) coinciden con las que presentan las aristas en una inmersión ortogonal. Esta analogía nos va a permitir en determinadas situaciones pasar de una representación a otra, como veremos más adelante.

La idea general ha sido establecer las relaciones entre representaciones topológicas y de visibilidad de grafos. En este sentido diremos que un grafo  $G$  es *bar-representable* si existe una configuración de segmentos verticales cuyo grafo de  $\varepsilon$ -visibilidad es precisamente  $G$ . En la Figura 0.19 se muestra un grafo representado topológicamente y una representación de bar-visibilidad que lo hace bar-representable.

Wismath [77], e independientemente Tamassia y Tollis [68] caracteri-

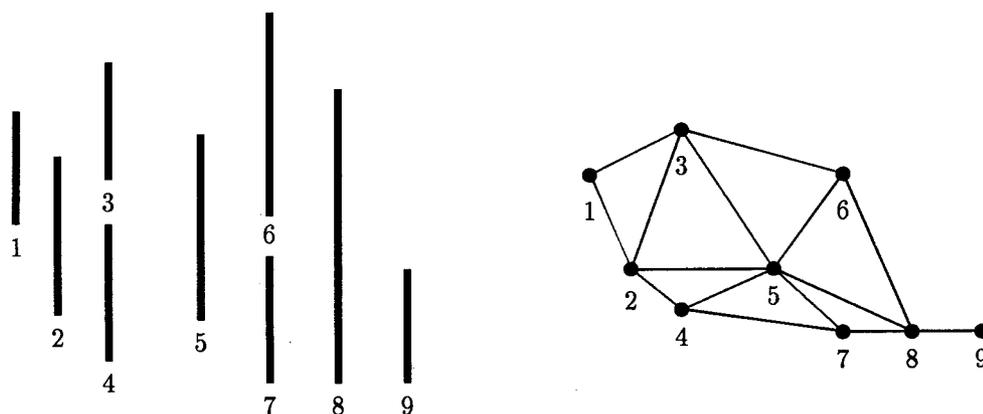


Figura 0.19: Las representaciones *bar* y *topológica* de un grafo.

zaron los grafos bar-representables en el plano. Fueron éstos últimos quienes plantearon el problema en superficies como el cilindro [71] y la esfera [69], generalizado posteriormente por Cobos [19] para superficies de mayor género. En el caso del cilindro, una bar-representación de un grafo está compuesta por segmentos sobre generatrices representando los vértices, y las aristas son rayos de visibilidad en la dirección de los círculos máximos. El algoritmo VISIB-2C [71] construye una bar-representación cilíndrica de un grafo plano 2-conexo en tiempo lineal en el número de vértices del grafo.

En la esfera, los vértices se representan mediante segmentos sobre paralelos y los rayos de visibilidad siguen la dirección de los meridianos. Tamassia y Tollis [69] caracterizan los grafos que admiten una representación de visibilidad en la esfera. Demuestran que es equivalente la bar-representabilidad de un grafo  $G$  en la esfera y la existencia de un *st*-grafo esférico  $G'$  siendo  $G$  isomorfo al grafo no dirigido que se obtiene de  $G'$  ignorando la dirección de sus aristas y eliminando los vértices  $s$  y  $t$ . En este contexto, un *st-grafo esférico* es un digrafo acíclico plano con exactamente una fuente (vértices sin aristas entrantes),  $s$ , y exactamente un sumidero (vértices sin aristas salientes),  $t$ .

## 0.4 Aspectos algorítmicos

Dado que algunos de los resultados que se obtienen en este trabajo son algoritmizables, debemos establecer, también en este aspecto, la notación que utilizaremos para los algoritmos que se formulen.

Al ser nuestro objetivo el de la claridad, en general nos expresaremos utilizando el lenguaje natural a la hora de describir un algoritmo, aunque también indiquemos esquemáticamente los pasos a seguir.

La mayor o menor bondad o *complejidad* de un algoritmo depende de dos factores, a saber:

- El *tiempo* que tarda en realizarse.
- El *espacio* (memoria) que ocupa.

El segundo factor es un problema subsanable (dentro de lo posible), pues basta con ampliar la memoria al tamaño necesario.

El primer factor es el que en realidad va a marcarnos las pautas por las que debe regirse el diseño de un algoritmo, ya que *el tiempo* es un problema que no tiene solución posible. El tiempo empleado en la ejecución de un algoritmo es la suma de los tiempos de las operaciones que han sido realizadas. En el campo del diseño y análisis algorítmico se suele expresar dicho tiempo módulo una constante multiplicativa, dando lugar a lo que se conoce como *efectividad o complejidad del algoritmo*. Así, la *complejidad* de un algoritmo es una función  $g(n)$  donde  $n$  es el tamaño de la entrada del problema, definida como el mayor tiempo empleado para resolver el problema con entradas de tamaño  $n$ .

La notación que daremos a continuación, denominada *notación asintótica*, debida a Bachmann [2] (1894), y popularizada desde su incorporación sistemática por Knuth [42] (1976), será la que utilicemos en la presente memoria para medir la efectividad de un algoritmo, donde todos los análisis se re-

fieren a su complejidad computacional correspondiente al peor caso que pueda presentarse:

$O(f(n))$  denota al conjunto de todas las funciones  $g(n)$  tales que existen una constante positiva  $C$  y un natural  $n_0$  de forma que  $|g(n)| \leq Cf(n)$  para todo  $n \geq n_0$ .

$\Omega(f(n))$  denota al conjunto de todas las funciones  $g(n)$  tales que existen una constante positiva  $C$  y un natural  $n_0$  de forma que  $g(n) \geq Cf(n)$  para todo  $n \geq n_0$ .

$\Theta(f(n))$  denota al conjunto de todas las funciones  $g(n)$  tales que existen dos constantes positivas  $C_1$  y  $C_2$  y un natural  $n_0$  de forma que  $C_1f(n) \leq g(n) \leq C_2f(n)$  para todo  $n \geq n_0$ .

$o(f(n))$  denota al conjunto de todas las funciones  $g(n)$  tales que para toda constante positiva  $C$  existe un natural  $n_0$  de forma que  $g(n) \leq Cf(n)$  para todo  $n \geq n_0$  (o, equivalentemente  $\lim_{n \rightarrow \infty} \frac{g(n)}{f(n)} = 0$ ).

Se deduce por tanto, que  $O(f(n))$  es el concepto que se utiliza para describir cotas superiores, mientras que  $\Omega(f(n))$  corresponde a cotas inferiores. El concepto que se necesita para algoritmos “óptimos” es  $\Theta(f(n))$ .

De esta forma, si el número de operaciones que realiza un algoritmo es

$$3n^4 + 20n^2 + 5n + 11 \leq (3 + 20 + 5 + 11)n^4 = 39n^4,$$

diremos que el algoritmo es  $O(n^4)$ .

Un algoritmo que sea  $O(n^p)$  con  $p \in \mathbf{N}$  se dice que es *polinomial*, utilizándose el término *lineal* para el caso  $p = 1$ . En el supuesto de que  $p = 0$  se dirá que el algoritmo *corre en tiempo constante*.

## 0.5 NP-completitud

El apelativo NP-*completo* ha sido uno de los pocos términos que en muy poco tiempo han ganado una notoriedad considerable. En el corto periodo de tiempo desde su introducción en los cercanos años 70, este término ha venido a simbolizar el abismo de la intratabilidad que los diseñadores de algoritmos tenían que afrontar cada vez con más frecuencia cuando pretendían resolver problemas más grandes y complejos. Una amplia variedad de problemas surgidos de investigaciones en áreas como matemáticas o computación se conocen actualmente como NP-completos, y la colección de tales problemas continúa aumentando diariamente.

La eficiencia de un algoritmo, como indicamos en la sección anterior, se mide en función del tiempo que tarda en resolver el problema. Sin embargo, esta consideración no es muy precisa puesto que las características del computador utilizado influyen notablemente; trabajando con la misma máquina podemos hablar de algoritmos más o menos eficientes.

Anteriormente introdujimos el concepto de algoritmo polinomial y en contraposición diremos que un algoritmo es *exponencial* si la función que define su complejidad no se puede acotar. Esta distinción adquiere importancia cuando se trabaja con tamaños grandes en las entradas. La Tabla 0.5 muestra las diferencias entre las efectividades de algoritmos de distinta complejidad en función del tamaño de la entrada. Notemos el excesivo crecimiento que se produce en algoritmos exponenciales.

tiempo /tamaño	10	100	500	1000
$1000n$	0.01 s.	0.1 s.	0.5 s.	1 s.
$100n \log n$	0.003 s.	0.06 s.	0.45 s.	1 s.
$n^3$	0.001 s.	1 s.	2 min.	16 min.
$2^n$	0.001 s.	40000 siglos		

Estos ejemplos indican algunas de las razones de porqué los algoritmos

polinomiales son, generalmente, preferibles a los exponenciales. Esta perspectiva y la distinción entre los dos tipos de algoritmos son básicos en el concepto de intratabilidad y Teoría de la NP-completitud, materia a la que está dedicada esta sección.

La distinción entre algoritmo polinomial y exponencial data de 1964, debida a Edmonds [22] y Cobham [18]. En particular, Edmonds catalogaba a los algoritmos polinomiales como “buenos” algoritmos y conjeturaba que ciertos problemas podrían no ser resueltos por tales algoritmos. En esta línea, un problema no estará “bien resuelto” hasta que se conozca un algoritmo polinomial que lo haga y un problema será *intratable* si es tan duro que, probablemente, ningún algoritmo polinomial pueda resolverlo.

Todas estas y muchas más consideraciones condujeron a Cook [20] en 1971 a la Teoría de la NP-completitud. En su trabajo “The Complexity of Theorem Proving Procedures” introdujo el concepto de “reducción polinomial” y definió la clase de problemas de decisión NP demostrando que la mayoría de los problemas catalogados hasta entonces de intratables, en su versión de problemas de decisión, pertenecían a dicha clase. En tercer lugar demostró que un problema en NP, el problema de “satisfacibilidad”, tenía la propiedad de que cualquier otro problema en NP se podía reducir polinomialmente a él. Como conclusión, el problema de satisfacibilidad era el “más duro” en NP y conjeturaba esta misma naturaleza para otros problemas en la clase. Fue Karp [41] quien los catalogó como problemas NP-completos presentando una colección de ellos.

Seguidamente damos todos estos conceptos y resultados básicos de NP-completitud. Podemos realizar un estudio más detallado de esta materia consultando el texto de Garey y Johnson [29] *Computers and Intractability: a guide to the Theory of NP-Completeness*.

Un *problema de decisión* es aquel cuyas dos únicas soluciones son SÍ y NO. Si  $\Pi$  es uno de estos problemas denotamos por  $D_{\Pi}$  al conjunto de sus entradas y por  $Y_{\Pi}$  al subconjunto de entradas con respuesta afirmativa.

Se dice que un algoritmo es *determinista* si para cada entrada del problema proporciona una solución. La *clase de problemas*  $P$  está formada por aquellos problemas de decisión para los que existe un algoritmo determinista polinomial que los resuelve.

Para introducir la segunda gran clase de problemas de decisión, la clase NP, necesitamos considerar otro tipo de algoritmos. Un *algoritmo no determinista* está compuesto por dos etapas: la etapa “guessing” y la etapa “checking”. Dada una entrada  $I$  de un problema de decisión, la etapa “guessing” origina una estructura  $S$ . Los datos  $I$  y  $S$  constituyen la entrada de la etapa “checking”, la cual comprueba de forma determinista si la estructura  $S$  hace que la entrada  $I$  pertenezca a  $Y_{\Pi}$ . Un algoritmo no determinista “resuelve” el problema de decisión  $\Pi$  si se tienen las dos propiedades siguientes para todas las entradas  $I \in D_{\Pi}$ :

- a) Si  $I \in Y_{\Pi}$ , entonces existe alguna estructura  $S$  que, una vez originada en la primera etapa a partir de  $I$ , conduce a la respuesta sí para  $I$  y  $S$  en la segunda etapa.
- b) Si  $I \notin Y_{\Pi}$ , entonces no existe ninguna estructura  $S$  que, originada para  $I$ , conduzca a la respuesta sí para  $I$  y  $S$ .

En realidad un algoritmo no determinista no resuelve el problema en el sentido estricto de la palabra, sino que “comprueba” si cierta estructura asociada a una entrada es una solución real a dicha entrada.

Un algoritmo no determinista que resuelve (comprueba) un problema de decisión  $\Pi$  se dice que corre en tiempo *polinomial* si existe un polinomio  $p$  tal que, para cada entrada  $I \in Y_{\Pi}$ , existe una estructura  $S$  que conduce a la respuesta sí para  $I$  y  $S$  en la segunda etapa en tiempo  $p$  en función del tamaño de  $I$ . Notemos que esta condición hace que exista una cota polinomial sobre el tamaño de  $S$  ya que solamente una cantidad polinomial de tiempo se puede emplear para leer la estructura  $S$  en la segunda etapa.

La *clase de problemas* NP está formada por aquellos problemas de decisión que pueden ser resueltos (comprobados) por algoritmos no deterministas polinomiales.

La relación entre las clases P y NP es fundamental en la Teoría de NP-completitud. Cualquier problema de decisión resoluble por un algoritmo determinista polinomial también es resoluble por uno no determinista polinomial, es decir  $P \subseteq NP$ . Para ver esto, solamente se necesita observar que cualquier algoritmo determinista puede constituir la etapa “checking” de un algoritmo no determinista. Si  $\Pi \in P$  y  $A$  es un algoritmo determinista polinomial para  $\Pi$ , se puede obtener uno no determinista polinomial para  $\Pi$  simplemente utilizando  $A$  como segunda etapa e ignorando la primera. Por tanto,  $\Pi \in P$  implica  $\Pi \in NP$ .

Por otra parte, existen fuertes razones para creer que esta inclusión es propia, es decir  $P \neq NP$ . Los conocimientos actuales conducen a trabajar bajo esta hipótesis en lugar de encaminar los esfuerzos para demostrar lo contrario. Una visión del mundo NP, tal y como lo consideramos, está mostrada en la Figura 0.20.

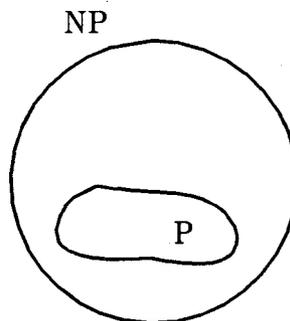


Figura 0.20: El mundo NP.

Bajo este supuesto, la distinción entre P y NP-P es significativa e importante. Todos los problemas en P se pueden resolver mediante algoritmos

polinomiales, mientras que todos los problemas en NP-P son intratables. Así, dado un problema de decisión  $\Pi \in \text{NP}$ , si  $P \neq \text{NP}$ , nos gustaría saber cuál de las dos posibilidades se tiene para  $\Pi$ . Por supuesto, hasta que no se pruebe que  $P \neq \text{NP}$ , no podemos esperar demostrar que un problema particular pertenece a NP-P. Por esta razón, la teoría de la NP-completitud esta enfocada a obtener resultados del tipo “si  $P \neq \text{NP}$  entonces  $\Pi \in \text{NP-P}$ .”

Dados dos problemas de decisión  $\Pi_1$  y  $\Pi_2$ , diremos que una función  $f : D_{\Pi_1} \rightarrow D_{\Pi_2}$  es una *transformación o reducción polinomial* de  $\Pi_1$  a  $\Pi_2$ , denotando  $\Pi_1 \infty \Pi_2$ , si satisface las dos condiciones siguientes:

- a)  $f$  es computable por un algoritmo polinomial,
- b)  $\forall I \in D_{\Pi_1}, I \in Y_{\Pi_1} \iff f(I) \in Y_{\Pi_2}$ .

La existencia de una transformación polinomial de  $\Pi_1$  a  $\Pi_2$  nos indica que si existe un algoritmo polinomial que resuelve  $\Pi_2$ , entonces existe otro que resuelve  $\Pi_1$ , esto es,  $\Pi_2 \in P \implies \Pi_1 \in P$ . Equivalentemente, si  $\Pi_1$  es intratable entonces  $\Pi_2$  también lo es.

Un problema de decisión  $\Pi$  es *NP-completo* si  $\Pi \in \text{NP}$  y, dado cualquier problema  $\Pi' \in \text{NP}$ ,  $\Pi' \infty \Pi$ . La observación anterior conduce a identificar los problemas NP-completos como “los más duros” en NP. Si un problema NP-completo se puede resolver en tiempo polinomial, entonces también son resolubles polinomialmente todos los problemas de la clase NP. Si un problema de NP es intratable entonces también lo son todos los NP-completos. La Figura 0.21, suponiendo  $P \neq \text{NP}$ , muestra una ampliación del mundo NP.

La definición de problema NP-completo dada anteriormente no proporciona una forma sencilla de demostrar que un problema pertenece a dicha clase, puesto que tendríamos que definir una transformación polinomial para cada problema de NP. La siguiente definición, equivalente a la anterior, sí constituye un método sencillo para probar la naturaleza NP-completa de un problema, a partir de otro conocido problema NP-completo. Un problema  $\Pi$

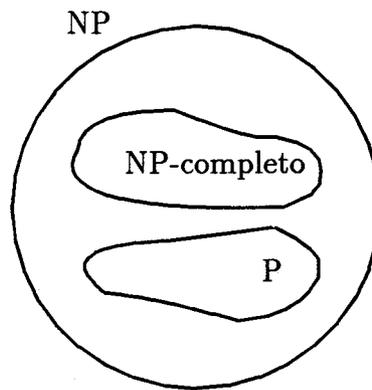


Figura 0.21 : El mundo NP ampliado.

es NP-*completo* si:

- a)  $\Pi \in \text{NP}$ , y
- b) existe un problema NP-completo  $\Pi'$  tal que  $\Pi' \in \Pi$ .

Por tanto, para demostrar que un problema  $\Pi$  es NP-completo nos basta con realizar los siguientes cuatro pasos:

- a) demostrar que  $\Pi$  pertenece a la clase NP,
- b) seleccionar un conocido problema NP-completo  $\Pi'$ ,
- c) construir una transformación  $f$  de  $\Pi'$  a  $\Pi$  y
- d) probar que  $f$  es una transformación polinomial.

Hasta ahora nuestro estudio se ha centrado en la clase de problemas NP, sin embargo pueden existir problemas fuera de esta clase que sean “tan

duros” como los NP-completos. A ellos dedicamos el concepto de problema “NP-duro” para problemas generales.

Un *problema de búsqueda* está formado por un conjunto de *entradas* y para cada entrada un conjunto de *soluciones*. Un algoritmo *resuelve* un problema de búsqueda si, dada una entrada, devuelve la respuesta NO cuando su conjunto de soluciones es vacío y en otro caso proporciona una solución contenida en dicho conjunto.

La generalización de “transformación polinomial” para problemas de búsqueda está motivada por la observación de que cualquier transformación polinomial entre dos problemas de decisión proporciona un algoritmo polinomial que resuelve el primero a partir de uno, también polinomial, que soluciona el segundo.

Dados dos problemas de búsqueda  $\Pi$  y  $\Pi'$ , una *reducción de Turing polinomial* de  $\Pi$  en  $\Pi'$  es un algoritmo  $A$  que resuelve  $\Pi$  utilizando una hipotética subrutina  $S$  para resolver  $\Pi'$  de forma que, si  $S$  fuera un algoritmo polinomial para  $\Pi'$ , entonces  $A$  también sería un algoritmo polinomial para  $\Pi$ . La existencia de esta reducción se denota  $\Pi \propto_T \Pi'$ .

Un problema de búsqueda  $\Pi$  es *NP-duro* si existe un problema  $L$  NP-completo tal que  $L \propto_T \Pi$ .

No es difícil observar que un problema NP-duro no se resuelve polinomialmente a menos que  $P=NP$ .

El primer problema NP-completo es un problema de decisión sobre variables booleanas, el problema de SATISFACIBILIDAD (Cook, 1971 [20]), y se enuncia en los siguientes términos.

Sea  $U = \{u_1, u_2, \dots, u_n\}$  un conjunto de *variables booleanas*. Una *asignación de verdad* para  $U$  es una función  $t : U \rightarrow \{T, F\}$ . Si  $t(u) = T$  decimos que  $u$  es *verdadera* bajo  $t$ ; si  $t(u) = F$  decimos que  $u$  es *falsa*. Si  $u$  es una variable de  $U$  entonces  $u$  y  $\bar{u}$  son *literales* sobre  $U$ . El literal  $u$  es verdadero

bajo  $t$  si y sólo si la variable  $u$  es verdadera bajo  $t$ ; el literal  $\bar{u}$  es verdadero si y sólo si la variable  $u$  es falsa.

Una *cláusula* sobre  $U$  es un conjunto de literales sobre  $U$ , por ejemplo  $\{u_1, \bar{u}_4, u_7\}$ ; representa la disyunción de dichos literales y se *satisface* bajo una asignación de verdad si, y sólo si, al menos uno de sus miembros es verdadero para la asignación. La cláusula dada se satisface bajo cualquier asignación de verdad  $t$  excepto cuando  $t(u_1) = F$ ,  $t(u_4) = T$  y  $t(u_7) = F$ . Una colección  $C$  de cláusulas sobre  $U$  se *satisface* si y sólo si existe alguna asignación de verdad para  $U$  que satisface simultáneamente todas las cláusulas de  $C$ . Tal asignación de verdad se llama *asignación de verdad satisfactoria* para  $C$ .

Bajo estos conceptos, el problema de SATISFACIBILIDAD se define como sigue:

#### Satisfacibilidad (SAT) [20]

*Entrada:* Un conjunto  $U$  de variables y una colección  $C$  de cláusulas sobre  $U$ .

*Pregunta:* ¿Existe una asignación de verdad satisfactoria para  $C$ ?

Por ejemplo,  $U = \{u_1, u_2\}$  y  $C = \{\{u_1, \bar{u}_2\}, \{\bar{u}_1, u_2\}\}$  constituyen una entrada de SAT cuya respuesta es “sí”. Una asignación de verdad satisfactoria para  $C$  vendría dada por  $t(u_1) = t(u_2) = T$ . Por otro lado, reemplazando  $C$  por  $C' = \{\{u_1, u_2\}, \{u_1, \bar{u}_2\}, \{\bar{u}_1\}\}$  obtenemos una entrada cuya respuesta es “no”;  $C'$  no se satisface.

A partir de este primer problema NP-completo fueron surgiendo diversos problemas de la misma naturaleza en áreas muy diversas. Como hemos indicado anteriormente, en [29] podemos encontrar una amplia lista de ellos, aunque no todos, por supuesto, ya que las investigaciones actuales han dado lugar a muchos más, no incluidos.

Presentamos en último lugar dos de dichos problemas NP-completos, variantes de SAT, a los que nos remitiremos en los últimos capítulos de esta memoria.

**3-satisfacibilidad (3SAT) [20]**

*Entrada:* Un conjunto  $U$  de variables y una colección  $C$  de cláusulas sobre  $U$  compuestas por 3 elementos.

*Pregunta:* ¿Existe una asignación de verdad satisfactoria para  $C$ ?

**Not-all-equal-3sat (ne3sat) [60]**

*Entrada:* Un conjunto  $U$  de variables y colección  $C$  de cláusulas sobre  $U$  compuestas por 3 elementos.

**PREGUNTA:** ¿Existe una asignación de verdad sobre  $U$  tal que cada cláusula de  $C$  contenga al menos un literal verdadero y uno falso?

# **Parte I**

## **Inmersiones ortogonales cilíndricas**

---

Existe una amplia literatura sobre inmersiones ortogonales planas, la cual se va ampliando últimamente con algunos trabajos en tres dimensiones. Es clara la aplicación que presentan este tipo de dibujos en el diseño de circuitos impresos, los cuales se realizan sobre placas que determinan superficies diversas, en general no planas. Es por ello por lo que consideramos este tipo de cuestiones en otras superficies, y comenzamos, por su simplicidad, con el cilindro.

Dos capítulos componen esta parte de la memoria. En el primero analizamos la estructura constituida por el conjunto de ángulos y codos asociado a una inmersión cilíndrica. Esta noción queda recogida bajo el concepto de asignación ortogonal y desarrollamos una teoría que nos conduce a determinar en tiempo  $O(n^2 \log n)$  una asignación ortogonal con el mínimo número de codos entre todas las asociadas a la inmersión cilíndrica dada.

Abordamos en el segundo capítulo la forma de traducir la información que nos proporciona una asignación ortogonal y pasar así a una inmersión ortogonal cilíndrica, para lo cual tenemos que determinar la dirección de los segmentos. Este paso en el plano es directo obteniéndose situaciones análogas para las dos direcciones posibles, horizontal y vertical, pero la estructura métrica del cilindro hace que la situación sea totalmente distinta en este caso, surgiendo diferencias importantes en cuanto a la asignación de dirección. Este análisis nos conduce a la noción de asignaciones ortogonales realizables, como aquellas que determinan una inmersión ortogonal cilíndrica. Diseñamos algoritmos lineales de construcción de asignaciones ortogonales realizables y caracterizamos las realizables entre todas las posibles asociadas a una inmersión cilíndrica dada. Entre ellas nos cuestionamos la determinación de la óptima en cuanto al número de codos, caracterizándola en términos teóricos.

El algoritmo GLOBAL construye en tiempo polinomial inmersiones ortogonales equivalentes que aproximan el número de codos.

Pasando a un enfoque local del número de codos en cada arista, demostramos que todo grafo 4-plano admite una inmersión ortogonal cilíndrica

con, a lo más, dos codos en cada arista. Se muestran ejemplos que prueban la condición mínima de este valor dos. El algoritmo LOCAL proporciona en tiempo lineal en el número de vértices de la inmersión, tales inmersiones ortogonales equivalentes, para grafos 2-conexos. Es de destacar que los resultados aquí obtenidos para el cilindro, mejoran los que se dan en el plano.

# Capítulo 1

## Asignaciones ortogonales cilíndricas

Toda inmersión ortogonal define de forma intrínseca un conjunto de ángulos y codos, obtenidos al recorrer las aristas de cada cara de la inmersión. Esta estructura posee un significado propio si se considera independiente del dibujo, pudiendo ser definida sobre cualquier inmersión cilíndrica, bajo ciertas restricciones para que defina una representación factible. Este hecho conduce al concepto de *asignación ortogonal* asociada a una inmersión cilíndrica. En este capítulo definimos este concepto y nos planteamos el problema de encontrar una asignación ortogonal asociada a una inmersión cilíndrica dada, que proporcione el mínimo número de codos. La resolución se basa en la transformación del problema en uno de flujos de mínimo coste sobre una red asociada a la inmersión. Esta transformación constituye una equivalencia entre ambos problemas, la cual da lugar a un algoritmo que construye una asignación ortogonal cilíndrica óptima en tiempo  $O(n^2 \log n)$  donde  $n$  es el número de vértices de la inmersión cilíndrica dada. El desarrollo de este capítulo es paralelo al trabajo realizado por Tamassia [64] sobre el problema análogo en el plano.

## 1.1 Introducción

La topología que presenta una placa en la que se ha diseñado un circuito, puede variar en función del fin destinado para ella. Si se requieren conexiones entre vértices situados en los bordes entonces, o bien los trazados tienen que intersectar a otros dibujados anteriormente, situación no deseable, o bien tienen que seguir la topología de una superficie cilíndrica. En general, un gran número de conexiones conduce a superficies de mayor género, problema que trataremos en la segunda parte de la memoria. Nos centramos en los dos capítulos siguientes en el cilindro como superficie para dibujos ortogonales de grafos.

Como aplicaciones de este estudio, además de la citada en el diseño de circuitos VLSI [46, 75], podemos mencionar el trazado de planos del suelo arquitectónico [49, 57] y el trazado estético de diagramas utilizados en diseño de sistemas de información [4, 3].

Estos problemas quedan modelizados de forma general en cuestiones sobre inmersiones ortogonales de grafos. Muchas variantes han surgido en este estudio, tratadas básicamente en un contexto plano. Tamassia [64] diseña un algoritmo polinomial que proporciona una inmersión ortogonal plana equivalente a una dada, con el mínimo número de codos entre todas las posibles. Este problema fue planteado en un trabajo de Storer [62], en el que la existencia de un algoritmo polinomial para resolverlo quedaba como problema abierto. Si en lugar de partir de una inmersión se parte únicamente de un grafo, el resultado cambia notoriamente. Garg y Tamassia [31] prueban que obtener una inmersión de un grafo con el mínimo número de codos es un problema de naturaleza NP-dura.

La condición necesaria y suficiente para que un grafo admita una inmersión ortogonal es que sea un grafo 4-plano. Este tipo de grafos han sido los considerados en los problemas anteriores. Si se reduce la valencia de los vértices, Di Battista, Liotta y Vargiu [7] obtienen un algoritmo polinomial

que construye inmersiones ortogonales con el mínimo número de codos para 3-grafos.

Nuestro trabajo sigue esta línea planteada en la superficie cilíndrica. También en este caso, la condición necesaria y suficiente para que un grafo cilíndrico admita una inmersión ortogonal en el cilindro es que sea un 4-grafo. La condición necesaria es obvia y la suficiente, aunque es simple, queda probada en el Lema 2.2.2, que veremos más adelante.

Comenzamos considerando la estructura que constituye los cambios de dirección y los codos existentes en una posible inmersión ortogonal cilíndrica, definiendo el concepto de *asignación ortogonal*, en el que no se fijan las direcciones de los segmentos. El paso a la malla cilíndrica se abordará en el Capítulo 2.

Partimos de una inmersión cilíndrica  $P$  de un 4-grafo, de forma que contenga al menos un ciclo esencial para que no constituya una inmersión plana. Las distintas asignaciones ortogonales asociadas a  $P$  representan las posibles formas de dibujar  $P$  en la malla cilíndrica conservando la inmersión. Nos planteamos la caracterización de la asignación ortogonal con el mínimo número de codos asociada a  $P$ , quedando determinada al transformar el problema en uno de flujo de mínimo coste. El algoritmo polinomial ASIG-OPTI construye una asignación ortogonal cilíndrica óptima asociada a  $P$ .

Aunque el problema ha sido planteado con 4-grafos, los resultados también son validos para multigrafos, conservando la condición sobre la valencia de los vértices.

Pasamos a dar unas definiciones previas antes de abordar de lleno el problema.

Un *cilindro (infinito)*  $C$  es la unión de una familia infinita de líneas paralelas a una recta, llamada *eje* del cilindro, y a la misma distancia de ella. Un cilindro se define también como el lugar geométrico de los puntos

que equidistan del eje. Alternativamente, se puede definir como la unión de una familia infinita de circunferencias del mismo radio, cuyos centros están situados sobre el eje y dibujadas sobre planos ortogonales a éste. Las rectas y las circunferencias definen de forma natural un sistema de coordenadas sobre el cilindro, puesto que cada punto de  $C$  se obtiene como la intersección de una única recta y una única circunferencia. De igual forma, estas dos familias de curvas definen una *mall*a sobre el cilindro; a las rectas paralelas se las conoce como *generatrices* y corresponden a la dirección vertical, mientras que las circunferencias perpendiculares a dichas rectas son los *círculos máximos* del cilindro, asociados a la dirección horizontal (ver Figura 1.1).

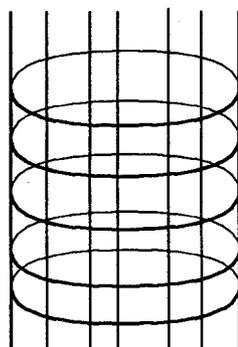


Figura 1.1: Malla en la superficie cilíndrica.

Un cilindro infinito es homeomorfo al plano menos un punto, o a la esfera menos dos, puntos que se conocen como *Polo Norte* y *Polo Sur*. Este homeomorfismo nos conduce a representar el cilindro como una banda limitada por rectas verticales identificadas entre sí, obtenida al desarrollar el cilindro por una de sus generatrices. Las geodésicas del cilindro pasan a ser rectas en su representación plana y la malla está constituida por una familia de rectas horizontales y otra de verticales. Las Figuras 1.2 y 1.3 ilustran estos hechos.

En el Capítulo 0 definimos inmersión de un grafo en cualquier superficie.

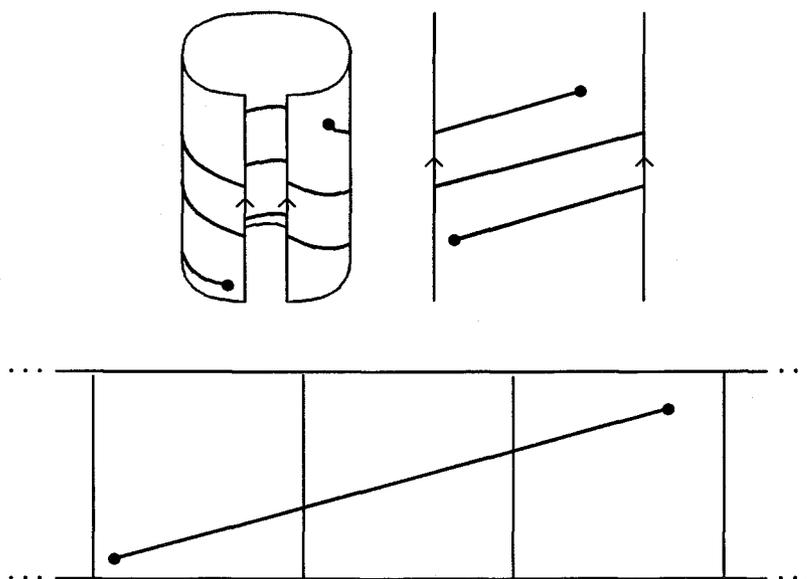


Figura 1.2: Representación plana del cilindro.

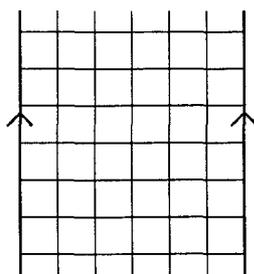


Figura 1.3: Malla en la representación plana del cilindro.

En este punto vamos a particularizarla al cilindro, por lo que una inmersión cilíndrica de un grafo estará constituida por puntos del cilindro representando los vértices, y las aristas serán segmentos que no se intersectan entre sí, excepto en sus extremos. Diremos que un grafo es *cilíndrico* si admite una inmersión en el cilindro. Es fácil ver que, al igual que grafo plano y esférico son conceptos equivalentes, grafo cilíndrico y plano también lo son. Los homeomorfismos existentes entre plano, cilindro y esfera hacen que si un grafo admite una inmersión en alguna de las tres superficies, también admita inmersiones en las otras dos. Esto es, los conceptos de grafo plano, cilíndrico y esférico son equivalentes.

De forma análoga, particularizamos la definición de inmersión ortogonal para la superficie cilíndrica. Una *inmersión en la malla del cilindro* o *inmersión ortogonal cilíndrica* de un grafo es una inmersión cilíndrica en la que las aristas están representadas por cadenas de segmentos consecutivos, horizontales y verticales. En este punto es importante señalar que al hablar de inmersiones en la malla del cilindro, estamos estableciendo la principal diferencia con el caso plano, ya que estamos tratando un problema de naturaleza métrica y no sólo topológica.

Las inmersiones cilíndricas, al igual que las demás, dan lugar a un sistema de rotación en los vértices y a un conjunto de ciclos esenciales. En este caso, un ciclo de la inmersión será *esencial* u *homotópicamente no nulo* si la curva asociada a dicho ciclo, la cual estará compuesta por segmentos, no encierra un disco abierto. Si una inmersión cilíndrica no contiene ningún ciclo esencial la llamaremos inmersión plana.

Toda inmersión cilíndrica determina un conjunto de caras sobre el cilindro, entre las que aparecen dos no acotadas (solamente una si es una inmersión plana), una de las cuales contiene al Polo Norte y la otra al Polo Sur. Si denotamos por  $n$ ,  $m$  y  $f$  al número de vértices, aristas y caras de la inmersión, respectivamente, entonces la fórmula de Euler establece que  $n - m + f = 2$ .

La estructura topológica de una inmersión cilíndrica se describe seña-

lando los ciclos esenciales y recorriendo las aristas que constituyen la frontera de cada cara, explicitando las caras no acotadas o externas. Cada cara  $f$  de la inmersión tendrá asociada una lista circular  $P(f)$  formada por las aristas que la forman, recorridas en el sentido de las agujas del reloj, es decir, dejando el interior de la cara a la derecha. Notemos que cada arista aparece exactamente en dos de estas listas, es decir pertenece a la frontera de dos caras de la inmersión. Si estas dos caras son la misma decimos que la arista constituye un *punte*.

Pasamos a considerar algunas definiciones básicas en teoría de flujos en redes, herramientas necesarias para nuestros resultados. Como referencia podemos citar [59] y [45].

Una *red*  $N = (U, A, s, t, b, c)$  viene definida por:

- a) Un conjunto finito de vértices  $U$ .
- b) Dos vértices distinguidos  $s$  y  $t$ , llamados *fuentes* y *sumidero*, respectivamente.
- c) Un conjunto  $A$  de pares ordenados de vértices, llamados *aristas*, que incluye la arista especial de retorno  $(t, s)$ .
- d) Una *capacidad* positiva  $b(a)$  para cada arista  $a$ .
- e) Un *coste*  $c(a)$  para cada arista  $a$ .

Un *flujo* en  $N$  es una función  $x : A \rightarrow R$  verificando:

- a)  $0 \leq x(a) \leq b(a)$  para cada arista  $a$ .
- b)  $\sum_u x(u, v) - \sum_w x(v, w) = 0$  para cada vértice  $v$

donde  $u$  y  $w$  son vértices de  $U$  de forma que  $(u, v)$  y  $(v, w)$  sean aristas de  $A$ , respectivamente.

El *valor* del flujo  $x$ , denotado  $|x|$ , viene dado por el flujo en la arista de retorno:

$$|x| = x(t, s).$$

El *coste* del flujo  $x$  es la cantidad:

$$\text{COST}(x) = \sum_{a \in A} c(a)x(a).$$

La Figura 1.5 muestra un flujo en la red descrita en la Figura 1.4.

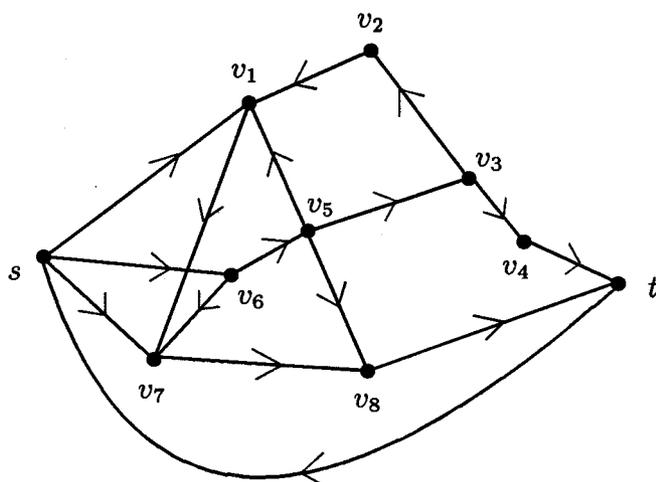
El problema del flujo de mínimo coste consiste en determinar el flujo de mínimo coste entre todos los de valor fijo dado. Se plantea de la siguiente forma:

MIN-COST-FLOW: Dada una red  $N$  y un número positivo  $z$ , encontrar un flujo  $x$  en  $N$  de valor  $|x| = z$  que tenga coste mínimo.

El *algoritmo de aumentación* para el problema MIN-COST-FLOW consiste en aumentar el flujo inicial nulo a lo largo de caminos de coste mínimo [45]. En el caso de capacidades enteras y costes no negativos, la obtención del camino más corto se puede realizar mediante una variación del algoritmo de Dijkstra. Por lo tanto, el algoritmo de aumentación es de orden  $O(z(|U| + |A|) \log |U|)$ .

Un *ciclo aumentador del flujo*  $C$  con respecto a un flujo  $x$  es un ciclo, con una asignación de recorrido, en el grafo no dirigido que resulta de ignorar las direcciones de las aristas de  $N$  y eliminar la arista de retorno  $(t, s)$ , verificando las siguientes propiedades:

- a) Para cada arista  $a$  de  $C$  cuya dirección en  $N$  coincida con el recorrido del ciclo, se tiene que  $x(a) < b(a)$ . A estas aristas se le llaman *directas*.
- b) Para cada arista  $a$  de  $C$  cuya dirección en  $N$  sea opuesta al recorrido del ciclo, se tiene que  $x(a) > 0$ . A estas aristas se le llaman *inversas*.



arista	capacidad	coste	arista	capacidad	coste
$(s, v_1)$	1	0	$(v_5, v_3)$	4	2
$(s, v_6)$	9	3	$(v_5, v_8)$	1	1
$(s, v_7)$	3	5	$(v_6, v_5)$	10	1
$(v_1, v_7)$	7	1	$(v_6, v_7)$	1	0
$(v_2, v_1)$	2	-1	$(v_7, v_8)$	6	-2
$(v_3, v_2)$	3	1	$(v_4, t)$	2	0
$(v_3, v_4)$	3	1	$(v_8, t)$	8	7
$(v_5, v_1)$	5	4	$(t, s)$	$\infty$	0

Figura 1.4: Ejemplo de red.

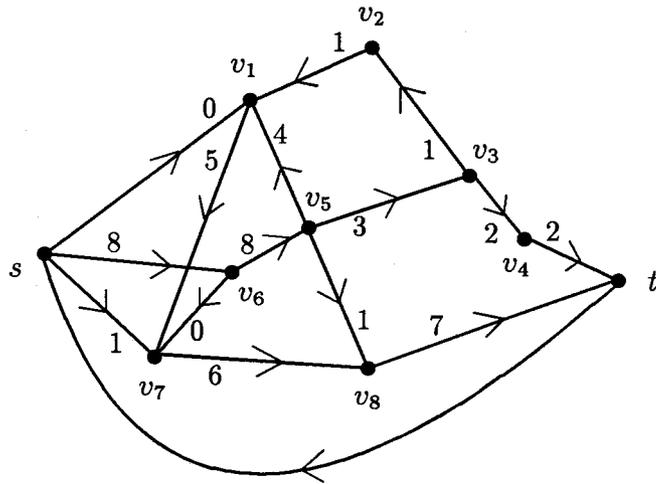


Figura 1.5: Flujo en la red de la Figura 1.4 de valor 9 y coste 55.

El *coste* de un ciclo aumentador del flujo viene dado por la suma de los costes de las aristas directas menos la suma de los costes de las aristas inversas.

En la Figura 1.6 se presentan dos ciclos aumentadores del flujo de la Figura 1.5, uno de coste negativo y otro positivo.

## 1.2 Asignaciones ortogonales de inmersiones cilíndricas

Con el objetivo general de obtener inmersiones ortogonales cilíndricas que presenten pocos codos, comenzamos introduciendo el concepto de asignación ortogonal, la cual vendrá a constituir el conjunto de codos y cambios de dirección (horizontal y vertical) factibles de las aristas de una inmersión cilíndrica, caracterizando aquella que proporciona la asignación ortogonal con el mínimo número de codos entre todas las posibles asociadas a una inmersión cilíndrica

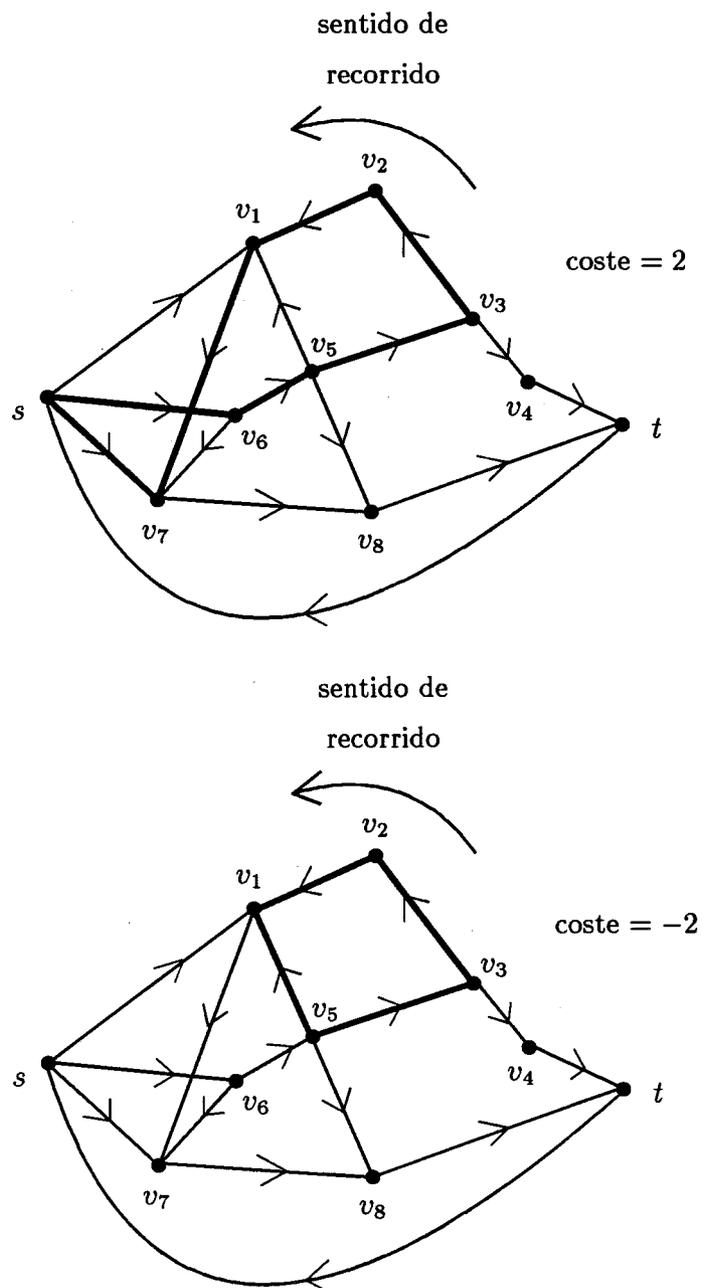


Figura 1.6: Dos ciclos aumentadores del flujo definidos sobre el mostrado en la Figura 1.5.

dada. De esta caracterización se obtiene un algoritmo polinomial para construirla, el algoritmo ASIG-OPTI, presentado al final del capítulo.

Se llama *asignación ortogonal*  $H$  de una inmersión cilíndrica  $P$  de un grafo 4-plano  $G$  a un conjunto de listas  $H = \cup_{f \in F} H(f)$  ordenadas circularmente, verificando las siguientes propiedades:

- a) Para cada cara  $f$  de  $P$  se define una lista  $H(f)$ , cuyos elementos tienen la forma  $r = (e[r], s[r], a[r])$ , donde:
  - a.1)  $e$  es una arista de  $f$ .
  - a.2)  $s$  es una secuencia binaria en la que cada dígito indica la existencia de un codo en  $e$ . Los valores 0 y 1 denotan ángulo menor y ángulo mayor respectivamente, formado por el codo en la cara  $f$ . Si  $e$  no contiene codos pondremos la notación  $\epsilon$ .
  - a.3)  $a$  toma el valor 1, 2, 3 ó 4. Este valor nos informa sobre el ángulo que forma en la cara  $f$  la arista  $e$  con la siguiente en la ordenación. Cada unidad en el valor de  $a$  representa un ángulo de 90 grados. El valor 4, es decir 360 grados, indica que la arista siguiente es ella misma, es decir  $e$  contiene un vértice de valencia 1.

Para definir  $H(f)$  recorreremos circularmente las aristas de  $f$ , dejando la cara a la derecha, y a cada una de ellas le asociamos los valores  $s$  y  $a$  de la forma anterior.

El número de codos  $C(H)$  de una inmersión ortogonal viene dado por

$$C(H) = \frac{1}{2} \sum_f \sum_{r \in H(f)} |s[r]| \quad (1.1)$$

donde  $|s|$  es la longitud de la secuencia binaria  $s$ .

- b) Sean  $r$  y  $r'$  un par de elementos de la asignación ortogonal asociados a la misma arista. Entonces la secuencia binaria  $s[r']$  se puede obtener de  $s[r]$  cambiando el orden de la secuencia binaria  $s[r]$  e intercambiando

ceros y unos. Esta propiedad significa que cada arista tiene que tener una descripción consistente en cada una de las caras en las que aparece.

c) Se define para cada elemento  $r$

$$\text{rotación}(r) = \text{ceros}(s[r]) - \text{unos}(s[r]) + (2 - a[r])$$

donde  $\text{ceros}(s[r])$  y  $\text{unos}(s[r])$  denotan el número de ceros y unos en la secuencia  $s$ , respectivamente. Se tiene que verificar:

$$\sum_{r \in H(f)} \text{rotación}(r) = \begin{cases} -4 & \text{si } f \text{ es cara externa con dos polos} \\ 0 & \text{si } f \text{ es cara externa con un polo} \\ 4 & \text{si } f \text{ es cara interna.} \end{cases}$$

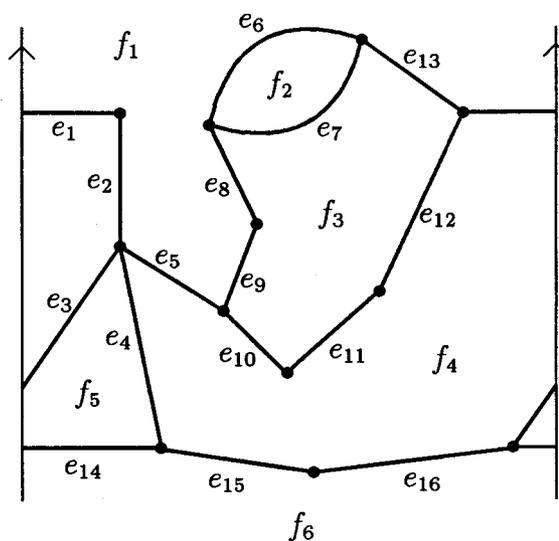
Esta propiedad significa que cada cara descrita por  $H$  es un polígono, es decir que las aristas que componen la cara forman un ciclo en la inmersión.

d) Para cada vértice  $v$  de  $G$  descrito por  $H$ , la suma de los ángulos entre pares de aristas de la misma cara, incidentes en  $v$ , los cuales vienen dados por el campo  $a$  de los elementos de  $H(f)$ , es igual a cuatro.

Se llama *segmento* de una asignación ortogonal a cualquiera de los siguientes elementos:

- Una arista sin codos.
- La parte de una arista comprendida entre un vértice y el primer codo de la secuencia correspondiente.
- La parte de una arista comprendida entre dos codos consecutivos.

La Figura 1.7 muestra una inmersión cilíndrica de un grafo 4-plano y una asignación ortogonal asociada a ella. Se explicita, para cada cara  $f_i$ ,  $i = 1, \dots, 6$ , su correspondiente lista  $H(f_i)$ . Se puede comprobar fácilmente que se verifican todas las propiedades de una asignación ortogonal.



$$H(f_1) = \{(e_1, \epsilon, 2), (e_{13}, \epsilon, 2), (e_6, 1, 1), (e_8, \epsilon, 3), (e_9, \epsilon, 1), (e_5, \epsilon, 1), (e_2, \epsilon, 3)\}$$

$$H(f_2) = \{(e_6, 0, 1), (e_7, 0, 1)\}$$

$$H(f_3) = \{(e_7, 1, 1), (e_{13}, \epsilon, 1), (e_{12}, 0, 1), (e_{11}, \epsilon, 3), (e_{10}, \epsilon, 1), (e_9, \epsilon, 1), (e_8, \epsilon, 2)\}$$

$$H(f_4) = \{(e_4, \epsilon, 1), (e_5, \epsilon, 2), (e_{10}, \epsilon, 1), (e_{11}, \epsilon, 3), (e_{12}, 1, 1), (e_1, \epsilon, 1), (e_2, \epsilon, 1), (e_3, 1, 1), (e_{16}, \epsilon, 2), (e_{15}, \epsilon, 1)\}$$

$$H(f_5) = \{(e_3, 0, 1), (e_4, \epsilon, 1), (e_{14}, \epsilon, 1)\}$$

$$H(f_6) = \{(e_{14}, \epsilon, 2), (e_{15}, \epsilon, 2), (e_{16}, \epsilon, 2)\}$$

Figura 1.7: Ejemplo de asignación ortogonal.

Los resultados que vamos a presentar a continuación hacen uso de técnicas de flujos en redes, por lo que comenzamos asociando una red a cada inmersión cilíndrica dada.

Sea  $G = (V, E)$  un grafo 4-plano y  $P$  una inmersión cilíndrica donde  $F$  es el conjunto de sus caras y  $P(f)$  es la lista circular asociada a la cara  $f \in F$  en  $P$ . Definimos la red  $N(P) = (U, A, s, t, b, c)$  asociada a  $G$  y  $P$ , de la siguiente forma.

- Sea  $\hat{V}$  el subconjunto de vértices de  $V$  que tienen valencia menor o igual que tres. Los vértices de  $N(P)$  los forman los vértices de  $\hat{V}$ , las caras de  $P$ , la fuente  $s$  y el sumidero  $t$ .

$$U = \hat{V} \cup F \cup \{s, t\}$$

- Las aristas de  $N(P)$  son de varios tipos

$$A = A_V \cup A_F \cup A_s \cup A_t \cup \{(t, s)\},$$

donde,

- $A_V$  está constituido por aristas del tipo  $(v, f)$ , donde  $v \in \hat{V}$  y  $f \in F$ , de forma que  $v$  es un extremo de alguna arista de  $P(f)$ . Las aristas de  $A_V$  tienen capacidad infinita y coste cero. Definimos

$$E(v, f) = \{e \in P(f) / \text{la arista } e \text{ y la siguiente en la lista } P(f) \text{ son incidentes en } v\}.$$

- $A_F$  contiene dos aristas  $(f, g)$  y  $(g, f)$  para cada par  $\{f, g\}$  de caras distintas, de forma que el conjunto

$$E(f, g) = \{e \in E / e \text{ aparece tanto en } P(f) \text{ como en } P(g)\}$$

es no vacío. Además,  $A_F$  contiene un lazo  $(f, f)$  por cada cara  $f$  que contenga un puente. Se define

$$E(f, f) = \{e \in E / e \text{ es un puente de la cara } f\}.$$

Las aristas de  $A_F$  tienen capacidad infinita y coste uno.

c)  $A_s$  contiene

c.1) Aristas del tipo  $(s, f)$  donde  $f$  es una cara interna que verifica  $|P(f)| \leq 3$ .

c.2) Aristas del tipo  $(s, v)$  donde  $v \in \hat{V}$ .

Las aristas de  $A_s$  tienen coste cero y capacidad:

$$b(s, f) = 4 - |P(f)| \quad \text{y} \quad b(s, v) = 4 - \delta(v)$$

d)  $A_t$  contiene aristas del tipo  $(f, t)$  donde  $f$  es una cara externa o una interna cuya lista  $P(f)$  tiene longitud  $|P(f)| \geq 5$ . Las aristas de  $A_t$  tienen coste cero y capacidad:

$$b(f, t) = \begin{cases} |P(f)| + 4 & \text{si } f \text{ es cara externa con dos polos} \\ |P(f)| & \text{si } f \text{ es cara externa con un polo} \\ |P(f)| - 4 & \text{si } f \text{ es una cara interna.} \end{cases}$$

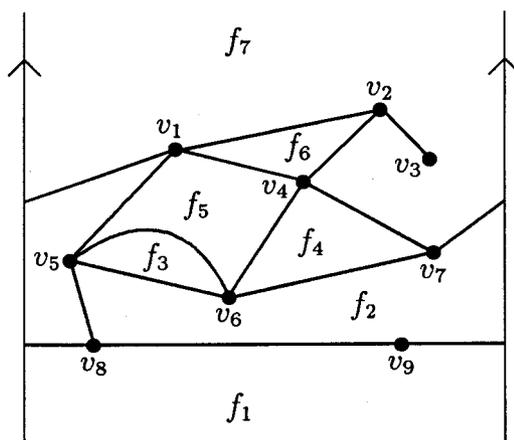


Figura 1.8: Inmersión cilíndrica.

La Figura 1.8 muestra una inmersión cilíndrica y su red asociada, distinguiendo los vértices de cada tipo (Figuras 1.9, 1.10 y 1.11).

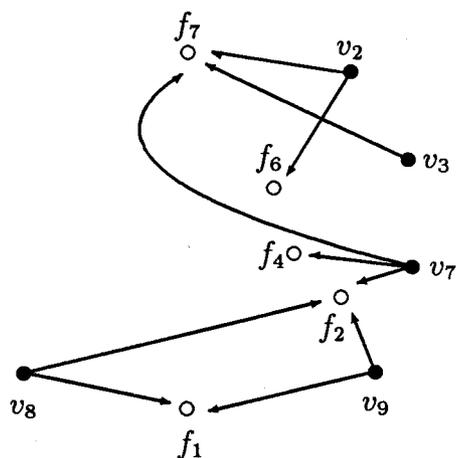


Figura 1.9: Aristas de  $A_V$ : capacidad =  $\infty$  y coste = 0.

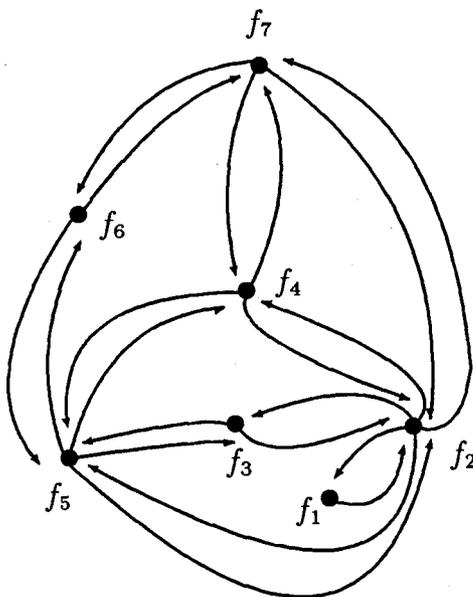


Figura 1.10: Aristas de  $A_F$ : capacidad =  $\infty$  y coste = 1.

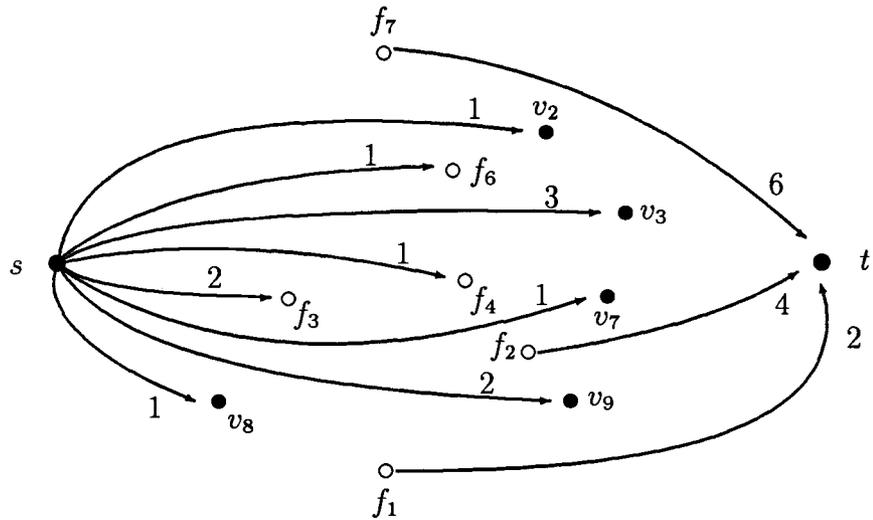


Figura 1.11: Aristas de  $A_s$  y  $A_t$ : la capacidad se muestra en cada arista y el coste es nulo.

Una vez definida la red  $N(P)$  asociada a la inmersión cilíndrica  $P$ , vamos a probar la equivalencia entre existencia de flujo en dicha red y asignación ortogonal asociada a  $P$ . Esta relación queda implícita al hablar de *flujo ortogonal* en  $N(P)$ , cuyo significado viene dado por las siguientes consideraciones:

- Cada unidad de flujo en la red  $N(P)$  representará un ángulo de 90 grados: para las aristas de  $A_V$  el flujo  $x(v, f)$  representa el ángulo que forman las aristas incidentes en  $v$  en la cara  $f$ , el cual viene dado por  $(x(v, f) + 1)90$ ; para las aristas de  $A_F$  el flujo  $x(f, g)$  representa el número de codos que aparecen a lo largo de las aristas que separan la cara  $f$  de la cara  $g$ , y que forman un ángulo de 90 grados con la cara  $f$ .
- La última propiedad en la definición de asignación ortogonal se traduce diciendo que la suma de los ángulos en cada vértice tiene que ser 360 grados. La propiedad anterior nos indicaba que cada cara tiene que ser un polígono.

c) El coste del flujo es igual al número de codos de la asignación ortogonal.

Vamos a imponer que el valor del flujo en  $N(P)$  sea igual a

$$z(P) = \sum_{u \in U} b(s, u) = \sum_{w \in U} b(w, t)$$

donde la igualdad entre los dos términos se obtiene de la fórmula de Euler como sigue. Definimos

$$F' = \{f \in F / f \text{ es cara interna y } |P(f)| \geq 5\}$$

$$F'' = \{f \in F / f \text{ es cara interna y } |P(f)| \leq 3\}$$

y, puesto que partimos de una inmersión cilíndrica  $P$  no plana, sean  $f_1$  y  $f_2$  las dos caras externas, conteniendo un Polo cada una de ellas. Entonces se tiene:

$$\begin{aligned} & \sum_{w \in U} b(w, t) - \sum_{u \in U} b(s, u) = \\ & |P(f_1)| + |P(f_2)| + \sum_{f \in F'} (|P(f)| - 4) - \left[ \sum_{f \in F''} (4 - |P(f)|) + \sum_{v \in V} (4 - \delta(v)) \right] = \\ & \sum_{f \in F} (|P(f)| - 4) + 8 + \sum_{v \in V} (\delta(v) - 4) = \\ & \sum_{f \in F} |P(f)| - 4|F| + 8 + \sum_{v \in V} \delta(v) - 4|V| = \\ & 2|E| - 4|F| + 8 + 2|E| - 4|V| = \\ & 4(|E| - |F| - |V| + 2) = 0. \end{aligned}$$

Este valor del flujo nos conduce a imponer que el flujo en las aristas de  $A_s$  y  $A_t$  sea igual a su capacidad. Además como a cada vértice de  $\hat{V}$  le llega exactamente una arista con capacidad a lo más tres, el flujo de las aristas de  $A_V$  no puede ser mayor que tres.

Los dos resultados siguientes establecen la equivalencia entre la existencia de flujo de valores enteros en  $N(P)$  de valor  $z(P)$  y asignación ortogonal asociada a  $P$  definida anteriormente.

**Lema 1.2.1.** *Asociado a cada asignación ortogonal de una inmersión cilíndrica  $P$  de un grafo 4-plano existe un flujo ortogonal  $x$  en  $N(P)$  con valores enteros, cuyo valor es  $z(P)$  y cuyo coste es igual al número de codos de  $P$ .*

**Demostración:** Sea  $H$  una asignación ortogonal de  $P$ . El flujo  $x$  en  $N(P)$  se construye a partir de  $H$  de la siguiente forma. Para cada arista  $(v, f)$  de  $A_V$ , el flujo  $x(v, f)$  se obtiene del campo  $a$  de los elementos de  $H(f)$  asociados a la arista  $(v, f)$ :

$$x(v, f) = \sum_{r \in R(v, f)} (a[r] - 1),$$

donde

$R(v, f) = \{r \in H(f) / \text{tanto } e[r] \text{ como la arista } e[r'] \text{ asociada al elemento } r' \text{ siguiente a } r \text{ en la lista } H(f) \text{ son incidentes en } v\}$ .

Para cada arista  $(f, g)$  de  $A_F$ , el flujo  $x(f, g)$  se obtiene del campo  $s$  de los elementos de  $H(f)$  asociados a la arista  $(f, g)$ :

$$x(f, g) = \sum_{r \in R(f, g)} \text{ceros}(s[r]),$$

donde

$$R(f, g) = \{r \in H(f) / e[r] \in E(f, g)\}.$$

El flujo  $x(f, g)$  representa los ángulos de 90 grados dentro de la cara  $f$  que aparecen a lo largo de las aristas que separan  $f$  de  $g$ . Si  $f = g$ , entonces  $x(f, f)$  es igual al número total de codos a lo largo de los puentes de  $f$ .

El flujo en las aristas de  $A_s$  y  $A_t$  es igual a la capacidad, como viene impuesto por el valor del flujo.

Para demostrar que la función  $x$  definida es un flujo, tenemos que probar por una parte que se mantiene la capacidad como cota superior del flujo, y por otro lado que el flujo se conserva en cada vértice.

La primera condición se puede comprobar fácilmente a partir de la definición de  $N(P)$  y  $x$ . La segunda condición se demuestra a partir de las propiedades de asignación ortogonal como sigue. Para los vértices de  $\hat{V}$ , y en virtud de la cuarta condición de una asignación ortogonal, se tiene

$$\begin{aligned} \sum_{f \in F} x(v, f) - x(s, v) &= \\ \sum_{f \in F} \sum_{r \in R(v, f)} (a[r] - 1) - (4 - \delta(v)) &= \\ \sum_{f \in F} \sum_{r \in R(v, f)} a[r] - \sum_{f \in F} \sum_{r \in R(v, f)} 1 - 4 + \delta(v) &= \\ 4 - \delta(v) - 4 + \delta(v) &= 0. \end{aligned}$$

Para probar la condición de flujo en los vértices de  $F$  notemos que

- $E(f, g) = E(g, f)$
- $\bigcup_{E(f, g) \neq \emptyset} R(f, g) = H(f)$
- $\bigcup_{E(v, f) \neq \emptyset} R(v, f) = H(f)$

Teniendo en cuenta estas consideraciones y a partir de las propiedades segunda y tercera en una asignación ortogonal, se tiene:

- a) Si  $f$  es una cara externa o una interna verificando  $|P(f)| \geq 5$ :

$$\begin{aligned} \sum_{u \in U} x(f, u) - \sum_{w \in U} x(w, f) &= \\ \sum_{E(f, g) \neq \emptyset} x(f, g) + x(f, t) - \left[ \sum_{E(v, f) \neq \emptyset} x(v, f) + \sum_{E(g, f) \neq \emptyset} x(g, f) \right] &= \\ \sum_{E(f, g) \neq \emptyset} \sum_{r \in R(f, g)} \text{ceros}(s[r]) + |P(f)| + c - & \end{aligned}$$

$$\begin{aligned}
& \left[ \sum_{E(v,f) \neq \emptyset} \sum_{r \in R(v,f)} (a[r] - 1) + \sum_{E(g,f) \neq \emptyset} \sum_{r \in R(g,f)} \text{ceros}(s[r]) \right] = \\
& \sum_{r \in H(f)} (\text{ceros}(s[r]) - \text{unos}(s[r]) - a[r] + 1) + |P(f)| + c = \\
& \sum_{r \in H(f)} (\text{rotación} - 1) + |P(f)| + c = \\
& -|P(f)| - c + |P(f)| + c = 0,
\end{aligned}$$

donde  $c$  es 4, 0, ó -4 si  $f$  es cara externa con dos polos, con un polo o cara interna, respectivamente.

b) Si  $f$  es una cara interna verificando  $|P(f)| \leq 3$ :

$$\begin{aligned}
& \sum_{u \in U} x(f, u) - \sum_{w \in U} x(w, f) = \\
& \sum_{E(f,g) \neq \emptyset} x(f, g) - \left[ \sum_{E(v,f) \neq \emptyset} x(v, f) + \sum_{E(g,f) \neq \emptyset} x(g, f) + x(s, f) \right] = \\
& \sum_{E(f,g) \neq \emptyset} \sum_{r \in R(f,g)} \text{ceros}(s[r]) - \\
& \left[ \sum_{E(v,f) \neq \emptyset} \sum_{r \in R(v,f)} (a[r] - 1) + \sum_{E(g,f) \neq \emptyset} \sum_{r \in R(g,f)} \text{ceros}(s[r]) + 4 - |P(f)| \right] = \\
& \sum_{r \in H(f)} (\text{ceros}(s[r]) - \text{unos}(s[r]) - a[r] + 1) + |P(f)| - 4 = \\
& \sum_{r \in H(f)} (\text{rotación} - 1) + |P(f)| - 4 = \\
& 4 - |P(f)| + |P(f)| - 4 = 0.
\end{aligned}$$

Por último, para los vértices  $s$  y  $t$ , la conservación del flujo se obtiene directamente de la definición del valor del flujo  $z(P)$ .

Para terminar tenemos que probar que el coste del flujo coincide con el número de codos de  $H$ , para lo que consideramos la expresión (1.1).

$$\begin{aligned}
COST(x) &= \sum_{a \in A} c(a)x(a) = \sum_{(f,g) \in A_F} x(f,g) = \\
&= \sum_{E(f,g) \neq \emptyset} \left( \sum_{r \in R(f,g)} ceros(s[r]) + \sum_{r \in R(g,f)} ceros(s[r]) \right) = \\
&= \sum_{E(f,g) \neq \emptyset} \sum_{r \in R(f,g)} ceros(s[r]) + unos(s[r]) = \\
&= \sum_{E(f,g) \neq \emptyset} \sum_{r \in R(f,g)} |s[r]| = \\
&= \frac{1}{2} \sum_f \sum_{r \in H(f)} |s[r]| = C(H).
\end{aligned}$$

■

Recíprocamente, el siguiente resultado nos proporciona una asignación ortogonal de  $P$  a partir de un flujo entero en  $N(P)$  de valor  $z(P)$  de forma que el coste de éste coincide con el número de codos de la asignación.

**Lema 1.2.2.** *Para cada flujo entero  $x$  en  $N(P)$  de valor  $z(P)$  existe una asignación ortogonal asociada a  $P$ , de forma que el número de codos que contiene es igual al coste del flujo  $x$ .*

**Demostración:** Los valores  $a$  y  $s$  asociados a las aristas de  $P$  en la asignación ortogonal que vamos a definir se obtienen a partir del flujo  $x$  en las aristas de  $A_V$  y  $A_F$ , respectivamente.

Para cada arista  $(v, f)$  de  $A_V$ , sean  $r_1, \dots, r_n$  los elementos de  $R(v, f)$  (ver la definición de  $R(v, f)$  en la prueba del Lema 1.2.1). Sea

$$a[r_1] = x(v, f) + 1$$

$$a[r_i] = 1 \text{ para } 2 \leq i \leq n.$$

Para cada par de aristas  $(f, g)$  y  $(g, f)$  en  $A_F$ , sean  $r'_1, \dots, r'_m$  los elementos de  $R(f, g)$ , y  $r''_1, \dots, r''_m$  los correspondientes elementos en  $R(g, f)$  (ver la definición de  $R(f, g)$  en la prueba del Lema 1.2.1). Si  $f = g$  entonces  $r'_i$  y  $r''_i$  son los pares de elementos en  $H(f)$  correspondientes a los dos lados del mismo puente. Sean

$$s[r'_1] = \begin{cases} 0^{x(f,g)} & \text{si } f = g \\ 0^{x(f,g)} 1^{x(g,f)} & \text{en otro caso,} \end{cases} \quad (1.2)$$

$$s[r''_1] = \begin{cases} 1^{x(g,f)} & \text{si } f = g \\ 0^{x(g,f)} 1^{x(f,g)} & \text{en otro caso,} \end{cases} \quad (1.3)$$

$$s[r'_i] = s[r''_i] = \epsilon \text{ para } 2 \leq i \leq m. \quad (1.4)$$

Veamos que los campos definidos anteriormente constituyen una asignación ortogonal asociada a  $P$ . La primera condición se verifica fácilmente teniendo en cuenta que la asignación ortogonal definida se ha obtenido a partir de  $P$ , y que  $a[r_1]$  se considera un valor en el conjunto  $\{1, 2, 3, 4\}$ . La segunda propiedad se sigue de las expresiones (1.2), (1.3) y (1.4), en las que se define el campo  $s$ . Para la tercera y cuarta condición, como veremos seguidamente, se tiene en cuenta que, como el valor del flujo es  $z(P)$ , el flujo en las aristas incidentes a la fuente o al sumidero coincide con la capacidad. De la conservación del flujo en los vértices de  $N(P)$  correspondientes a las caras de la inmersión, se tiene que

$$\begin{aligned} & \sum_{E(v,f) \neq \emptyset} x(v, f) + \sum_{E(f,g) \neq \emptyset} (x(g, f) - x(f, g)) = \\ & = \begin{cases} |P(f)| + 4 & \text{si } f \text{ es cara externa que contiene dos polos} \\ |P(f)| & \text{si } f \text{ es cara externa que contiene un polo} \\ |P(f)| - 4 & \text{si } f \text{ es cara interna.} \end{cases} \end{aligned}$$

Entonces, la tercera condición, rotación de los elementos de  $f$ , se deduce como sigue:

$$\sum_{r \in H(f)} \text{rotación}(r) = \sum_{r \in H(f)} [\text{ceros}(s[r]) - \text{unos}(s[r]) + (2 - a[r])] =$$

$$\begin{aligned}
& \sum_{E(f,g) \neq \emptyset} (x(f,g) - x(g,f)) + 2|P(f)| - \sum_{E(v,f) \neq \emptyset} x(v,f) - |P(f)| = \\
& \sum_{E(f,g) \neq \emptyset} (x(f,g) - x(g,f)) - \sum_{E(v,f) \neq \emptyset} x(v,f) + |P(f)| = \\
& = \begin{cases} -4 & \text{si } f \text{ es cara externa que contiene dos polos} \\ 0 & \text{si } f \text{ es cara externa que contiene un polo} \\ 4 & \text{si } f \text{ es cara interna.} \end{cases}
\end{aligned}$$

La propiedad cuarta se verifica en virtud de la conservación del flujo en los vértices del grafo:

$$\begin{aligned}
\sum_u x(u,v) - \sum_w x(v,w) &= x(s,v) - \sum_{E(v,f) \neq \emptyset} x(v,f) = \\
4 - \delta(v) - \sum_{E(v,f) \neq \emptyset} x(v,f) &= 0,
\end{aligned}$$

luego

$$\sum_{E(v,f) \neq \emptyset} x(v,f) = 4 - \delta(v),$$

por lo que

$$\begin{aligned}
\sum_{r \in R(v,f)} a[r] &= \sum_{E(v,f) \neq \emptyset} \sum_{i=1}^{n_f} a[r_i] = \sum_{E(v,f) \neq \emptyset} [x(v,f) + 1 + n_f - 1] = \\
\sum_{E(v,f) \neq \emptyset} x(v,f) + \delta(v) &= 4 - \delta(v) + \delta(v) = 4.
\end{aligned}$$

Finalmente, a partir de (1.1), (1.2), (1.3) y (1.4) se tiene la igualdad entre el coste del flujo y el número de codos de la asignación, de la siguiente forma:

$$\begin{aligned}
C(H) &= \frac{1}{2} \sum_f \sum_{r \in H(f)} |s[r]| = \frac{1}{2} \sum_{(f,g) \in A_F} (x(f,g) + x(g,f)) = \\
& \sum_{a \in A} c(a)x(a) = COST(x).
\end{aligned}$$

■

Por tanto, podemos hablar indistintamente de asignación ortogonal asociada a  $P$  o de flujo en  $N(P)$ .

Aunque la definición de asignación ortogonal no es breve, su significado sí debe ser claro. Una asignación ortogonal de una inmersión cilíndrica nos indica sobre los cambios de dirección en la malla que se producen en las aristas de  $P$ , bien sea al pasar de una arista a otra, incidentes en un vértice, o bien al considerar un codo en la arista. Se asegura que estas operaciones son compatibles entre caras con frontera común y que realmente se describe una inmersión cilíndrica. Notemos también que la asignación ortogonal es adimensional, es decir no fija la longitud de los segmentos sobre la malla, se conocen los cambios de dirección que se producen aunque no la dirección que presenta cada segmento. Si nos fijamos en la Figura 1.7, las aristas que delimitan la cara externa  $f_6$  presentan la misma dirección en la asignación, puesto que cualesquiera dos de ellas forman un ángulo 2 (180 grados) en dicha cara. Por el contrario, al pasar en  $f_4$  de  $e_1$  a  $e_2$  se forma un ángulo 1 (90 grados) en ella, luego se produce un cambio de dirección, y de igual forma,  $e_8$  y  $e_9$  presentan direcciones distintas al formar un ángulo 3 (270 grados) en  $f_1$ . La existencia de un codo origina un cambio de dirección en los segmentos que lo forman. Por ejemplo, la arista  $e_{12}$  está compuesta por dos segmentos formando un codo, luego tienen que tener direcciones distintas. En él, el ángulo 1 (270 grados) corresponde a la cara  $f_4$  y es la cara  $f_3$  la que contiene el de 90 grados o ángulo 0.

Sin embargo, aunque estemos hablando de direcciones en la malla del cilindro, no toda asignación ortogonal determina una inmersión ortogonal cilíndrica de  $P$ , como veremos en el Capítulo 2.

Continuamos con nuestro propósito de obtener la asignación ortogonal de  $P$  con el mínimo número de codos y para ello, los dos lemas anteriores y los resultados que damos a continuación nos conducen a una transformación de dicho problema en el MIN-COST-FLOW.

**Lema 1.2.3.** *Existe un flujo de mínimo coste en  $N(P)$  con valor  $z(P)$  que toma valores enteros.*

**Demostración:** Puesto que las capacidades en  $N(P)$  son valores enteros, este lema se sigue de forma inmediata a partir de los resultados de la teoría de flujos en redes [45]. ■

Este resultado y los Lemas 1.2.1 y 1.2.2 garantizan el siguiente teorema.

**Teorema 1.2.4.** *Dada una inmersión cilíndrica  $P$ , el mínimo número de codos entre los que contienen las asignaciones ortogonales asociadas a  $P$  es igual al coste mínimo de un flujo factible en  $N(P)$ .*

En este sentido podemos hablar de asignación ortogonal *óptima* y de flujo *óptimo* como aquellas estructuras que nos proporcionan el mínimo número de codos y coste mínimo, respectivamente.

La transformación en el problema MIN-COST-FLOW proporciona una sencilla caracterización de las asignaciones ortogonales óptimas, a partir de la de los flujos óptimos. La recogemos en los siguientes resultados:

**Lema 1.2.5.** *Un flujo  $x$  en la red  $N$  es de mínimo coste (óptimo) entre todos los flujos del mismo valor si y sólo si  $N$  no contiene ciclos aumentadores del flujo (con respecto a  $x$ ) de coste negativo.*

**Demostración:** Ver [45]. ■

**Teorema 1.2.6.** *Una asignación ortogonal  $H$  tiene el mínimo número de codos (óptima) entre todas las asignaciones ortogonales asociadas a una misma inmersión cilíndrica  $P$  si y sólo si no existe una curva dirigida, cerrada y simple  $J$  verificando las siguientes condiciones:*

- a)  $J$  intersecta a cada arista de  $H$  a lo más dos veces,
- b)  $J$  entra a vértices y codos de  $H$  cuyos campos  $a$  y  $s$  son mayores o iguales a 2, o iguales a 1, respectivamente.
- c) Más de la mitad de los cruces entre  $J$  y las aristas de  $P$  se producen en aristas que forman un codo de asignación 1, ángulo mayor, en la cara por la que entra la curva.

Además, si existe una curva  $J$  con estas características, se puede encontrar una asignación ortogonal  $H'$  asociada a  $P$  tal que

- a)  $H'$  tiene menos codos que  $H$ ;
- b)  $H$  y  $H'$  solamente se diferencian en las aristas y vértices cruzados por  $J$ .

**Demostración:** La prueba consiste en demostrar que existe una curva  $J$  en  $H$  verificando las propiedades anteriores si y sólo si la red  $N(P)$  admite un ciclo aumentador del flujo de coste negativo con respecto a  $x$ , siendo  $x$  el flujo ortogonal correspondiente a  $H$  que nos proporciona el Lema 1.2.1. Una vez probada esta afirmación, la primera parte del teorema se obtiene en virtud del Lema 1.2.5.

Supongamos que la red  $N(P)$  admite un ciclo aumentador del flujo de coste negativo  $C = (a_1, \dots, a_k)$ . Si  $C$  contiene a la fuente o al sumidero entonces contiene a dos aristas incidentes en uno de estos vértices, una de las cuales tiene que ser directa, sin embargo esto no es posible puesto que en dichas aristas  $x(a) = b(a)$ .

Construimos a partir de  $C$  una curva  $J$ , compuesta a trozos por los arcos  $J_1, \dots, J_k$ , de la siguiente forma:

- a) Marcamos un punto  $p_f$  en cada cara  $f$  de  $P$ .

- b) Por cada arista  $a_i$  de  $C$  dibujamos un arco  $J_i$  entre los extremos de  $a_i$ , es decir,
- b.1) si  $a_i = (v, f) \in A_V$  es directa, entonces dibujamos  $J_i$  desde  $v$  a  $p_f$  dentro de la cara  $f$ ,
  - b.2) si  $a_i = (v, f) \in A_V$  es inversa, entonces dibujamos  $J_i$  desde  $p_f$  a  $v$  dentro de la cara  $f$ ,
  - b.3) si  $a_i = (f, g) \in A_F$  entonces dibujamos  $J_i$ , o bien de  $p_f$  a  $p_g$ , o bien de  $p_g$  a  $p_f$  según si  $a_i$  es directa o inversa, respectivamente, atravesando una única arista de las compartidas por  $f$  y  $g$  en  $P$  y siguiendo la topología de la asignación. En el primer caso imponemos que  $J_i$  no atraviese ningún codo, en caso de que existan. En el último caso  $x(a_i) > 0$  y este valor representa el número de codos que presenta la arista, de forma que el ángulo mayor de los codos, denotado 1 en la asignación ortogonal, aparece en la cara  $g$ . En estas condiciones hacemos que  $J_i$  atraviese uno de estos codos por su ángulo mayor.

La primera condición de  $J$  se verifica puesto que la curva solamente atraviesa aristas de  $G$  correspondientes a las de  $C$  en  $A_F$ , y cada arista de  $G$  está asociada a lo más a dos de  $A_F$ . Por definición, si  $J$  entra por un codo lo hace por el ángulo de asignación 1 en  $H$ . Si entra por un vértice  $v$  lo hace desde una cara  $f$ , luego la arista  $(v, f)$  es inversa en  $C$  y su flujo es no nulo. Este flujo corresponde a un ángulo de asignación  $x(v, f) + 1$  (ver Lema 1.2.1), luego su campo  $a$  asociado es mayor o igual que 2. Por tanto se cumple la segunda condición del teorema. La tercera y última condición se sigue del hecho de que  $C$  es de coste negativo: en  $N(P)$  el coste de las aristas de  $A_V$  es cero y para las de  $A_F$  es uno, luego el coste de  $C$  lo determinan las aristas  $(f, g)$  y como éste es negativo el ciclo contiene más aristas inversas que directas. Por definición de  $J$  estas aristas inversas dan lugar a arcos de curva que atraviesan codos entrando por el ángulo mayor.

Para probar el recíproco supongamos que existe una curva  $J$  verificando

las propiedades  $a$ ,  $b$  y  $c$  del teorema. En primer lugar dividimos la curva en arcos  $J_1, \dots, J_k$ , donde cada  $J_i$  tiene un extremo en una cara, y el otro extremo es, o bien un vértice, o bien una cara: en el primer caso  $J_i$  no puede atravesar ninguna arista de  $P$ , mientras que en el último caso tiene que atravesar exactamente una arista. Construimos entonces, un ciclo aumentador del flujo  $C = (a_1, \dots, a_k)$  de la siguiente forma:

- a) si  $J_i$  va desde una cara  $f$  a un vértice  $v \in \hat{V}$ , sea  $a_i = (v, f)$ ;  $a_i$  es una arista inversa de  $C$ , puesto que, de la condición  $b$ , la arista  $(v, f)$  tiene flujo  $x(v, f) \geq 1$ ;
- b) si  $J_i$  va desde un vértice  $v \in \hat{V}$  a una cara  $f$ , sea  $a_i = (v, f)$ ;  $a_i$  es una arista directa de  $C$ ;
- c) si  $J_i$  va desde una cara  $f$  a una cara  $g$ , entonces sea  $a_i = (g, f)$  una arista inversa si  $J_i$  entra desde  $f$  por un codo de asignación 1; en otro caso sea  $a_i = (f, g)$  una arista directa de  $C$ .

Las condiciones  $a$  y  $b$  aseguran que el ciclo así definido es un ciclo aumentador del flujo. La condición  $c$ , junto con el coste nulo de las aristas del tipo  $(v, f)$ , asegura que  $C$  tiene coste negativo.

Para la segunda parte, la existencia de la curva  $J$  nos conduce a un ciclo aumentador del flujo de coste negativo, de forma que al aplicarlo, es decir, aumentando una unidad el flujo de las aristas directas y disminuyendo una unidad en las inversas, se obtiene otro flujo del mismo valor y menor coste. Este nuevo flujo, en virtud del Lema 1.2.2, asegura la existencia de una asignación ortogonal  $H'$  asociada a  $P$  verificando las condiciones expuestas.

■

En virtud de este teorema podremos hablar indistintamente de ciclo aumentador del flujo de coste negativo en el flujo o de su curva asociada en la asignación ortogonal. En este sentido diremos que un ciclo aumentador del

flujo de coste negativo es *homotópicamente nulo o esencial* si su curva asociada lo es.

Dado un ciclo aumentador del flujo de coste negativo en un flujo ortogonal  $x$ , es posible definir otro flujo  $x'$  con las propiedades vistas en el Teorema 1.2.6. Queremos dar unas ideas básicas de cómo varían los campos  $a$  y  $s$  de las asignaciones ortogonales asociadas a los flujos  $x$  y  $x'$ , basándonos en los Lemas 1.2.1 y 1.2.2. El campo  $a$  viene dado por el flujo en las aristas de la forma  $(v, f)$ . Si  $C$  contiene una arista  $(v, f)$ , la anterior o la siguiente será  $(v, g)$ , siendo una directa y otra inversa. Supongamos que  $(v, f)$  es directa y que  $(v, g)$  es inversa, es decir  $J$  entra en  $v$  desde la cara  $g$  y sale a la cara  $f$ . El flujo en  $(v, f)$  aumenta una unidad y el de  $(v, g)$  disminuye en una unidad; esto se traduce en un aumento del campo  $a$  de  $(v, f)$  y una disminución en el de  $(v, g)$ , es decir el ángulo por el que  $J$  entra a un vértice decrece una unidad, o lo que es lo mismo, aumenta una unidad el ángulo por el que sale. El flujo en las aristas de la forma  $(f, g)$  nos informa sobre el número de codos existentes en las aristas que separan  $f$  de  $g$ , que en la inmersión ortogonal viene dado por el campo  $s$ . Una curva que cruce una arista por un codo de  $H$  (por definición  $J$  entra por el ángulo mayor del codo) nos indica que en  $C$  aparece una arista inversa de la forma  $(f, g)$ , luego se disminuye el flujo en una unidad, es decir, desaparece dicho codo. Si el cruce no se produce en un codo significa que se trata de una arista directa, luego el aumentar el flujo se traduce en añadir un nuevo codo. Nuestro objetivo es reducir codos de la asignación ortogonal luego deben aparecer en el ciclo más aristas inversas que directas del tipo  $(f, g)$ , lo que se traduce en  $N(P)$  a ciclos de coste negativo. En la Figura 1.12 damos ejemplos de curvas en las condiciones de teorema anterior aplicadas a asignaciones ortogonales.

Como resultado inmediato del Teorema 1.2.6 obtenemos el siguiente corolario, ilustrado en la Figura 1.13(a).

**Corolario 1.2.7.** *Toda asignación ortogonal con el mínimo número de codos carece de aristas con codos de distinta asignación en la misma cara.*

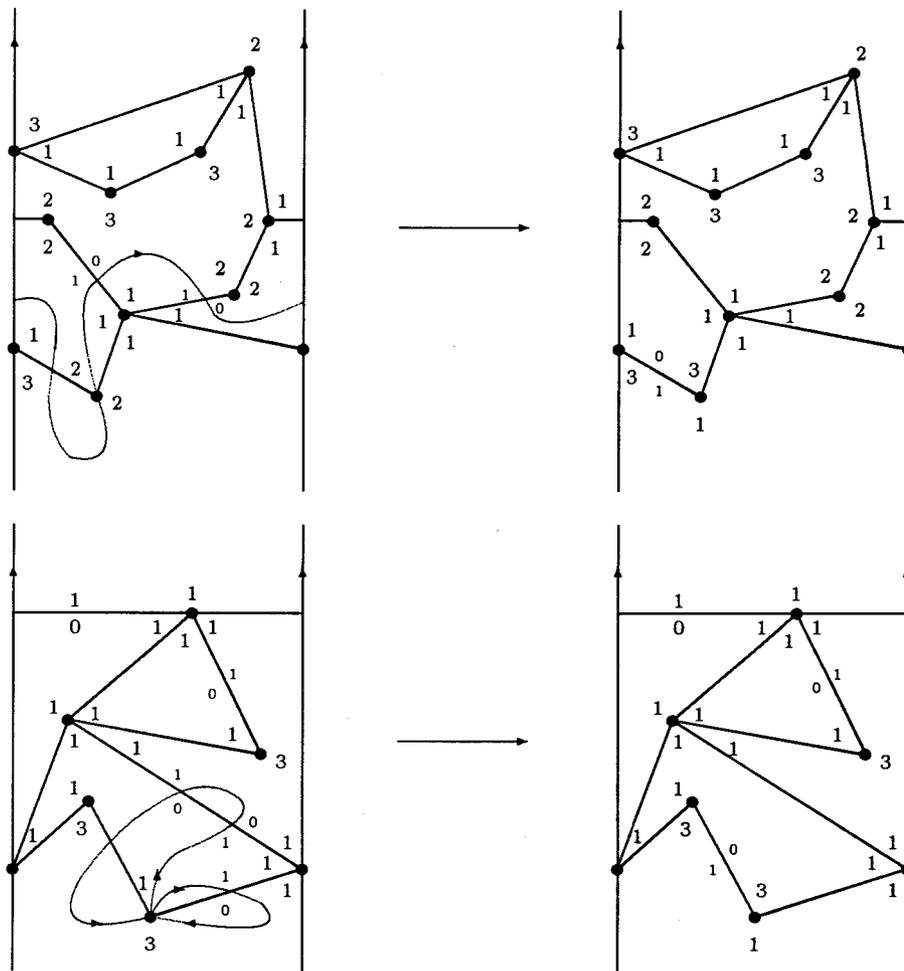


Figura 1.12: Aplicación de curvas a inmersiones ortogonales.

Este resultado también es válido en el plano, sin embargo, al contrario de lo que ocurre en [64], no podemos afirmar que todo puente esté representado en una asignación ortogonal óptima como una arista sin codos. En la Figura 1.13(b) se muestra un ejemplo.

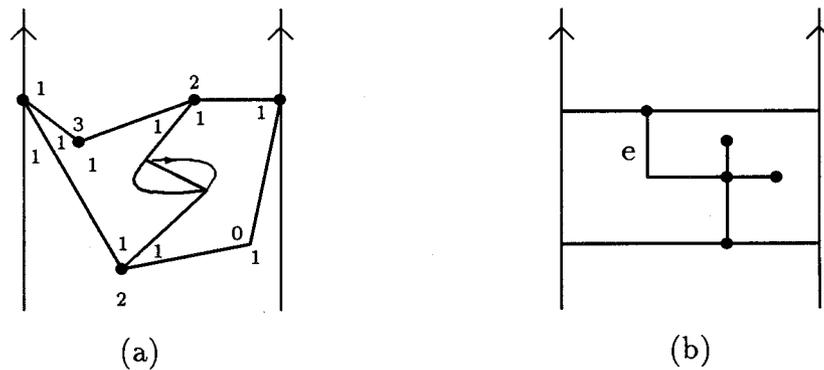


Figura 1.13: (a) La curva señalada elimina los dos codos de la arista; (b) la arista  $e$  es un puente con un codo.

Como conclusión, hemos demostrado que encontrar una asignación ortogonal óptima asociada a una inmersión cilíndrica 4-plana es equivalente a obtener un flujo óptimo en  $N(P)$ . Este hecho nos va a conducir a la obtención de un algoritmo eficiente para determinar dicha asignación óptima.

### 1.2.1 Aspectos algorítmicos

Dedicamos esta parte al diseño de un algoritmo polinomial que proporcione una asignación ortogonal óptima asociada a una inmersión cilíndrica 4-plana dada, basándonos en la caracterización demostrada anteriormente y transformación en un flujo óptimo en  $N(P)$ . Este flujo óptimo se puede obtener aplicando convenientemente el algoritmo de aumentación, el cual requiere tiempo polinomial en el número de vértices y aristas de la red y del valor impuesto al

flujo. El siguiente lema demuestra que estos valores en nuestro planteamiento crecen linealmente en el número de vértices de  $P$ .

**Lema 1.2.8.** *Sea  $P$  una inmersión cilíndrica de un grafo 4-plano  $G = (V, E)$ . Se verifican las siguientes desigualdades para la red  $N(P) = (U, A, s, t, b, c)$  y el valor del flujo  $z(P)$ :*

$$a) |U| \leq \frac{3}{2}|V| + 4$$

$$b) |A| \leq \frac{15}{2}|V| + 3$$

$$c) z(P) \leq 4|V|.$$

**Demostración:** De la definición de  $N(P)$  tenemos que

$$U = \hat{V} \cup F \cup \{s, t\}$$

y

$$A = A_V \cup A_F \cup A_s \cup A_t \cup \{(t, s)\}.$$

Recordamos que  $\hat{V}$  era el conjunto de vértices de  $G$  de valencia no mayor que 3. Sea  $V_k$  el conjunto de vértices de  $G$  de valencia  $k$ , entonces aplicando la fórmula de Euler se tiene que

$$|U| = |E| - |V_4| + 4.$$

Puesto que

$$2|E| = \sum_{k=1}^4 k|V_k|, \tag{1.5}$$

obtenemos

$$|U| = \frac{1}{2}(|V_1| + 2|V_2| + 3|V_3| + 2|V_4|) \leq \frac{3}{2}|V| + 4.$$

Para la segunda desigualdad tenemos que

$$|A| = |A_V| + |A_F| + |A_s| + |A_t| + 1 \leq$$

$$\leq 2|E| - 4|V_4| + 2|E| + |V| - |V_4| + |F| - |F_4| + 1$$

donde

$$F_k = \{f \in F / f \text{ es interna y } |P(f)| = k\}.$$

Aplicando de nuevo la fórmula de Euler y la igualdad 1.5 obtenemos

$$\begin{aligned} |A| &\leq 5|E| - 5|V_4| - |F_4| + 3 \leq \\ &\leq \frac{5}{2}(|V_1| + 2|V_2| + 3|V_3| + 4|V_4|) - 5|F_4| + 3 \leq \\ &\leq \frac{15}{2}|V| + 3. \end{aligned}$$

Recordemos que el valor impuesto al flujo es  $z(P) = \sum_{w \in U} b(w, t)$ . Puesto que consideramos inmersiones cilíndricas que no sean planas, existirán dos caras externas  $f_1$  y  $f_2$  conteniendo cada una de ellas un polo, por lo que nos queda

$$z(P) = |P(f_1)| + |P(f_2)| + \sum_{f \neq f_1, f_2} \max\{0, |P(f)| - 4\}.$$

Si definimos

$$F_k = \{f \in F / f \text{ es interna y } |P(f)| = k\},$$

entonces

$$\begin{aligned} z(P) &= |P(f_1)| + |P(f_2)| + \sum_{k=1}^{2|E|} |F_k|(k-4) - \sum_{k=1}^3 |F_k|(k-4) = \\ &= |P(f_1)| + |P(f_2)| + \sum_{k=1}^{2|E|} k|F_k| - 4 \sum_{k=1}^{2|E|} |F_k| + \sum_{k=1}^3 |F_k|(4-k). \end{aligned}$$

En los tres primeros sumandos se consideran las longitudes de todas las caras, esto es  $2|E|$ , y como el sumatorio del cuarto término está extendido solamente a las caras internas se tiene

$$z(P) = 2|E| - 4(|F| - 2) + 4 \sum_{k=1}^3 - \sum_{k=1}^3 k|F_k|,$$

donde el último término es negativo, por lo tanto

$$z(P) \leq 2|E| - 4(|F| - 2) + 4(|F| - 2) = 2|E|,$$

y como el grafo considerado es 4-plano  $|E| \leq 2|V|$ , por lo que concluimos que

$$z(P) \leq 4|V|.$$

■

El siguiente algoritmo proporciona una asignación ortogonal con el mínimo número de codos.

#### Algoritmo ASIG-OPTI

**Entrada:** Grafo 4-plano  $G$  e inmersión cilíndrica  $P$  de  $G$ .

**Salida:** Asignación ortogonal  $H$  asociada a  $P$  con el mínimo número de codos.

**Paso 1.** El algoritmo recibe como entrada a la inmersión cilíndrica  $P$ .

**Paso 2.** Se construye la red  $N(P)$ .

**Paso 3.** El algoritmo de aumentación proporciona el flujo de mínimo coste  $x$  en  $N(P)$  de valor  $z(P)$ .

**Paso 4.** A partir del flujo óptimo  $x$  se construye la asignación ortogonal óptima  $H$  correspondiente a  $x$  según indica el Lema 1.2.2.

La corrección de este algoritmo viene dada por el siguiente teorema:

**Teorema 1.2.9.** *Dado un grafo 4-plano  $G$  de  $n$  vértices y una inmersión cilíndrica  $P$  de  $G$ , el algoritmo ASIG-OPTI construye una asignación ortogonal de  $P$  con el mínimo número de codos en tiempo  $O(n^2 \log n)$ .*

**Demostración:** Siendo  $U$  y  $A$  los conjuntos de vértices y aristas de  $G$ , respectivamente, se puede comprobar fácilmente que los pasos 1 y 2 del algoritmo se pueden realizar en tiempo  $O(|U| + |A|)$ . Puesto que las capacidades en la red son enteras y los costes positivos, el paso 3 se puede ejecutar en tiempo  $O(z(P)(|U| + |A|) \log |U|)$ . El Lema 1.2.8 nos proporciona cotas superiores de  $|U|$ ,  $|A|$  y  $z(P)$ , las cuales son lineales en el número de vértices del grafo por lo que obtenemos el resultado requerido. ■

Encaminados a la obtención de inmersiones ortogonales cilíndricas con pocos codos, o en el caso óptimo con el mínimo número de codos, el algoritmo ASIG-OPTI construye una asignación ortogonal óptima de  $P$  y a partir de ella quisieramos llegar a una inmersión ortogonal cilíndrica equivalente a  $P$ .

Como hemos comentado anteriormente una asignación ortogonal no fija las direcciones de los segmentos; tendríamos dos posibilidades de horizontal o vertical para un segmento y éstas determinarían las direcciones de los restantes siguiendo la asignación. Sin embargo, existen asignaciones ortogonales para las que ninguna de las dos posibilidades conducen a inmersiones en la malla. Este hecho nos encamina a un estudio más detallado de la naturaleza de las asignaciones ortogonales, que realizaremos en el siguiente capítulo.

### 1.3 Conclusiones y problemas abiertos

Con este capítulo como origen, hemos abierto la línea de investigación sobre inmersiones ortogonales en superficies no planas. Por ser la “siguiente” al plano hemos comenzado con el cilindro. Con un trabajo de Tamassia como antecedente, hemos resuelto el problema de caracterizar y determinar en tiempo polinomial una asignación ortogonal con el mínimo número de codos, asociada a una inmersión cilíndrica dada  $P$ . Las técnicas empleadas son las mismas que presenta Tamassia para el caso plano, recurriendo a problemas de flujos de mínimo coste en redes.

Una vez caracterizado el conjunto de ángulos y codos óptimo asociado a  $P$ , surge de manera natural la necesidad de su conversión a una inmersión ortogonal equivalente a  $P$ . Para ello, puesto que una asignación ortogonal es adimensional y sin dirección, lo primero que habría que hacer es dotar de dirección a los segmentos. En [64] podemos observar que, en un contexto plano, este paso es automático, tratando indistintamente las dos direcciones de horizontal y vertical para un segmento. Hemos dedicado el Capítulo 2 al estudio de este tema, puesto que unos primeros ejemplos determinan la naturaleza totalmente distinta de la versión cilíndrica del problema.

En general, una asignación ortogonal óptima, aunque presente el mínimo número de codos, puede contener aristas con un número elevado de ellos, lo que hace antiestética la inmersión. De esta dificultad surge el planteamiento del problema aproximando el número de codos en cada arista por separado, que será tratado en el próximo capítulo.

Si en lugar de partir de una inmersión cilíndrica, solamente tenemos un grafo 4-plano como entrada, el problema sugiere ser bastante diferente. Garg y Tamassia [31] demuestran la naturaleza NP-completa del problema de decidir si un 4-grafo admite una inmersión rectilínea plana. Queda abierto el correspondiente en el caso cilíndrico para asignaciones ortogonales, en el que considerar todas las asociadas a cada inmersión del grafo dado nos conduciría a un método no efectivo. Habría que recurrir a nuevas técnicas de ataque, bien para obtener algoritmos polinomiales, o bien para demostrar su intratabilidad.

## Capítulo 2

# Inmersiones ortogonales cilíndricas

Bajo dos enfoques distintos consideramos el estudio de las inmersiones ortogonales cilíndricas en este capítulo. Desde un punto de vista global, obtenemos un método lineal para construir inmersiones en la malla con  $4n$  codos, y siguiendo la línea de minimizar el número de codos, introducimos el concepto de asignación ortogonal realizable como aquella asociada a una inmersión ortogonal. Caracterizamos, por un lado, las asignaciones ortogonales realizables, proporcionando un algoritmo lineal para obtener la correspondiente inmersión en la malla, y por otro lado, la que presenta el mínimo número de codos entre todas las realizables. Conservando la inmersión dada, se presenta un algoritmo polinomial para construir inmersiones ortogonales cilíndricas que aproximan el número óptimo de codos.

El otro punto de vista es local, estudiando inmersiones ortogonales cilíndricas con pocos codos en cada arista. Así, un grafo *k-ortogonal* es aquél que admite una inmersión cilíndrica cuyas aristas contienen como máximo  $k$  codos. Demostramos que  $k = 2$  es el mínimo valor óptimo para el cual todo 4-grafo cilíndrico es *k-ortogonal*, y damos un algoritmo lineal que, partiendo

como dato de entrada de una inmersión de un grafo 4-plano 2-conexo, construye inmersiones 2-ortogonales cilíndricas equivalentes a la dada. El valor  $k = 2$  constituye una mejora con respecto a los resultados existentes en el plano [52].

## 2.1 Introducción

Para poder realizar el trazado de un circuito sobre una placa, además de los datos acerca de los ángulos entre aristas y codos de éstas, se necesita conocer la dirección que presenta cada segmento del dibujo en la malla. La primera información viene dada por la asignación ortogonal, sin embargo, como comentamos anteriormente, ésta carece del segundo dato. Se presentan dos posibilidades en cuanto a la dirección de un segmento, vertical u horizontal, que en un contexto plano conducen a dos situaciones equivalentes. Ejemplos muy simples muestran que la asignación de vertical u horizontal a un segmento es muy significativa, puesto que en función de la dirección tomada, la asignación ortogonal puede o no conducir a una inmersión en la malla cilíndrica.

En este hecho radica la gran diferencia entre inmersiones planas y cilíndricas, que nos conduce a un tratamiento distinto del problema, distinguiendo entre asignaciones ortogonales realizables (aquellas que admiten una dirección sobre sus segmentos que determina una inmersión en la malla) y no realizables. En el caso plano todas las asignaciones son realizables para ambas direcciones.

Bajo esta dicotomía planteamos el problema de determinar la asignación ortogonal realizable que presente el mínimo número de codos entre todas las asociadas a la misma inmersión cilíndrica. De esta forma, el enfoque bajo el que se trabaja es global en la inmersión. En la Universidad de Roma "La Sapienza" se ha implementado el algoritmo obtenido por Tamassia [64] para determinar una inmersión ortogonal plana con el mínimo número de codos, conservando la inmersión del grafo 4-plano dado. La práctica demuestra

que, a medida que se aumenta el número de vértices de la inmersión, se originan aristas, en general en la cara externa, presentando un número elevado de codos. Aunque el valor total de codos sea mínimo en la representación, esto no evita la existencia de estas líneas de conexión “problemáticas” que dificultan la corriente en el circuito. Si los codos no están uniformemente repartidos en el trazado, la placa presentará zonas de diferente calidad.

Este es el motivo que ha conducido a los investigadores al diseño de placas con pocos codos en cada línea de conexión. La modelización de esta situación se realiza bajo el problema de determinar inmersiones ortogonales que presenten en cada arista un número de codos menor a una cierta cantidad. Diversos trabajos realizados desde este planteamiento podemos encontrar en un contexto plano, de los que citamos algunos de ellos. Storer [63] y Tamassia y Tollis [67] obtienen inmersiones ortogonales planas de grafos en una malla de tamaño  $n \times n$  con  $2n + 4$  codos si son 2-conexos, y  $2.4n + 2$  en otro caso. El número de codos a lo largo de cada arista es cuatro, como máximo. Para grafos 3-planos y 4-planos 3-conexos se han obtenido mejores cotas [40, 50, 51]. Biedl y Kant [12] presentan un algoritmo lineal que dibuja inmersiones ortogonales de grafos conexos en una malla de tamaño  $n \times n$  con, a lo más,  $2n + 2$  codos, donde cada arista se dobla como máximo dos veces. Más recientemente, Liu, Morgana y Simeone [53] obtienen un algoritmo lineal para representar ortogonalmente grafos 2-conexos con dos codos por arista, como máximo (excepto el octaedro). El número total de codos es  $2n + 4$  en una malla de tamaño  $(n + 1) \times (n + 1)$ .

Todos los trabajos mencionados anteriormente se enmarcan en un contexto plano, bien partiendo de un grafo, o bien de una inmersión, con el fin de obtener representaciones ortogonales. Siguiendo el objetivo general de este trabajo de generalización a superficies no planas, en este capítulo desarrollamos este tipo de problemas en la superficie cilíndrica. Los dos aspectos del problema, global y local, determinan dos partes diferenciadas en él. En una primera parte, tratamos sobre asignaciones ortogonales realizables, es decir aquellas que conducen a inmersiones ortogonales cilíndricas. Presentamos su caracterización en términos de inmersiones esféricas ascendentes de digrafos,

presentando, a partir de éstas, un algoritmo lineal que dibuja la asignación ortogonal realizable en la malla.

Dentro de la línea global y partiendo de una inmersión cilíndrica de un grafo 4-plano, obtenemos un algoritmo lineal que dibuja dicha inmersión en la malla con  $4n$  codos (se obtiene también cuatro codos, a lo más, en cada arista). En términos de flujos en redes, caracterizamos la asignación ortogonal realizable que presenta el mínimo número de codos, entre las asociadas a la inmersión dada. Para obtener algoritmos efectivos, proponemos una vía alternativa, en la que se presenta un algoritmo de orden  $O(n^2 \log n)$  que, conservando la inmersión cilíndrica dada, construye inmersiones ortogonales que aproximan el número óptimo de codos.

Pasando a la segunda parte, nos centramos en el enfoque local. En este caso partimos de un grafo y determinamos el valor mínimo de  $k$ ,  $k = 2$ , para el que todo grafo admite una inmersión cilíndrica ortogonal con, a lo más,  $k$  codos por arista. En último lugar se da un algoritmo que, para inmersiones de grafos 2-conexos, construye inmersiones ortogonales equivalentes en las condiciones anteriores. Es interesante observar que, en general, se mejoran con estos algoritmos los obtenidos para el plano [53].

## 2.2 Asignaciones ortogonales realizables

En esta sección se introduce el concepto de asignación ortogonal realizable como aquella asociada a una inmersión en la malla. Puesto que son estas inmersiones las que nos ocupan, caracterizamos tales asignaciones ortogonales y diseñamos un algoritmo lineal que construye sus representaciones ortogonales asociadas. Con el fin de distinguir las realizables y las no realizables, llevamos a cabo un estudio exhaustivo de ellas obteniendo asignaciones ortogonales realizables de inmersiones cilíndricas con  $4n$  codos. En la misma línea del capítulo anterior, nuestro objetivo consiste en determinar una asignación ortogonal realizable óptima, que reduzca, en general, los  $4n$  codos obtenidos.

La caracterización teórica de estas asignaciones realizables se determina en términos de flujos en redes, considerando flujos ortogonales realizables.

Es fácil observar que toda inmersión ortogonal cilíndrica lleva asociada intrínsecamente una asignación ortogonal, que se obtiene a partir de los ángulos que aquella define en la malla. El paso de un círculo máximo a una generatriz, o viceversa, determina un valor de 1 ó 3 en el campo  $a$ , dependiendo de la cara que se esté recorriendo. Un valor 2 en este campo significa que no se cambia de dirección. Los codos que presente la inmersión dan lugar a los campos  $s$  de las aristas. El hecho de partir de una inmersión ortogonal hace que los cambios de dirección que contiene verifiquen todas las condiciones para constituir una verdadera asignación ortogonal, o lo que es lo mismo un flujo ortogonal en la red. Recíprocamente, si pretendemos pasar de una asignación ortogonal a una inmersión en la malla lo primero que tenemos que hacer es dotar de dirección a los segmentos, sin embargo esta acción nos puede conducir a estructuras no factibles. Un sencillo ejemplo se considera en la Figura 2.1, en la que se muestra una asignación ortogonal de la inmersión representada  $P$ .

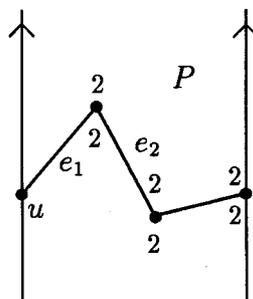


Figura 2.1: Asignación ortogonal  $H_1$  de la inmersión cilíndrica  $P$ .

Si se le asigna la dirección vertical a la arista  $e_1$ , la asignación determina la dirección también vertical del resto de las aristas, lo que conduce a un dibujo no factible, puesto que los vértices  $u_1$  y  $u_2$  tienen que ser los mismos, e iguales

a  $u$  (ver Figura 2.2 (a)). Por el contrario si  $e_1$  toma la dirección horizontal, el

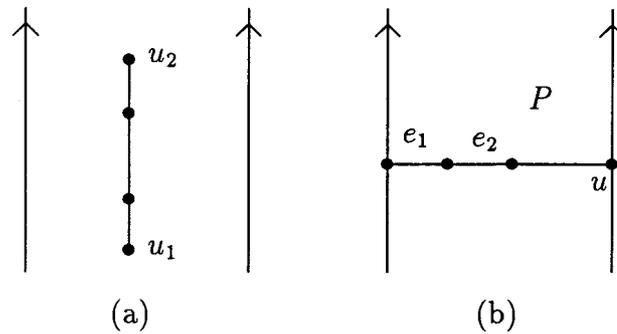


Figura 2.2: (a) La dirección vertical de  $e_1$  conduce a una inmersión no equivalente a  $P$ . (b) La dirección horizontal de  $e_1$  determina una inmersión ortogonal de  $P$ .

resto de las aristas también son horizontales, originando la inmersión ortogonal equivalente a  $P$  de la Figura 2.2 (b).

Puesto que en el primer caso obtenemos otra inmersión distinta de la dada, no podemos catalogar de ninguna forma dicha asignación y dirección en el contexto de nuestro problema. Este caso se presenta únicamente cuando  $P$  contiene un ciclo esencial y la asignación ortogonal considerada hace que sus aristas presenten la misma dirección. Si ésta es horizontal no tenemos problema, pero el caso vertical da lugar a un dibujo no factible.

En la Figura 2.3 se muestra otra asignación ortogonal de  $P$ , en la que la dirección horizontal para  $e_1$  da lugar a un dibujo en la malla en el que los vértices  $u_1$  y  $u_2$ , que deberían ser el mismo, se encuentran a diferente altura (Figura 2.4 (a)). Para salvar esta situación una de las aristas tiene que dejar de ser un segmento en la malla (Figura 2.4 (b)).

Si dotamos a la arista  $e_1$  de dirección vertical obtenemos la misma situación (ver Figura 2.5).

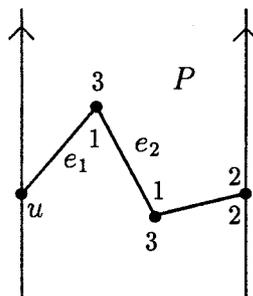


Figura 2.3: Asignación ortogonal  $H_2$  de la inmersión cilíndrica  $P$ .

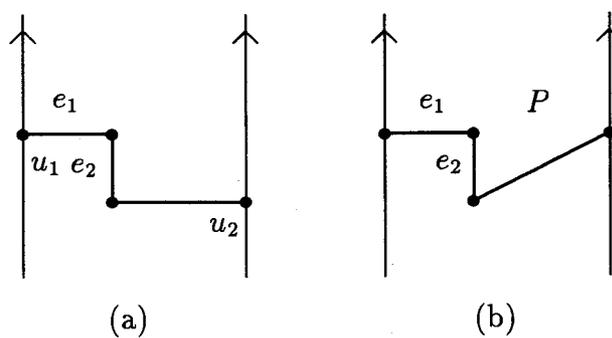


Figura 2.4: (a) Los vértices  $u_1$  y  $u_2$ , representando al vértice  $u$ , se encuentran a distinta altura. (b) Inmersión  $P$  no ortogonal.

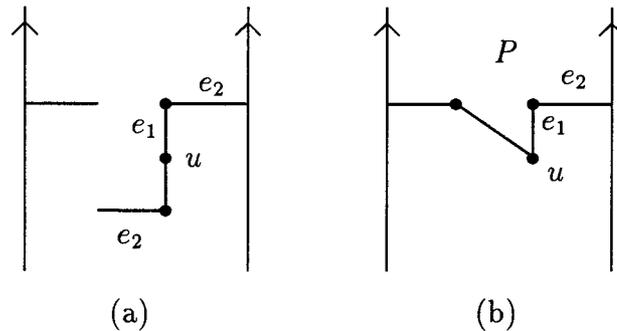


Figura 2.5: (a) La arista  $e_2$  contiene puntos a distinta altura. (b) Inmersión  $P$  no ortogonal.

Tamassia en [64] demuestra que para inmersiones planas de grafos 4-planos la relación entre asignación ortogonal e inmersión en la malla es biunívoca. Puesto que en el cilindro, como hemos visto, no se establece tal tipo de relación, surge de forma natural el problema de caracterizar los flujos ortogonales que originan una inmersión en la malla del cilindro, introduciendo el concepto de *asignación o flujo ortogonal realizable*.

Una asignación (flujo) ortogonal  $H$  de una inmersión cilíndrica  $P$  se dice *realizable* si existe una inmersión ortogonal equivalente a  $P$  cuyos ángulos y codos constituyen  $H$ .

En estos términos la asignación  $H_1$  es realizable, mientras que  $H_2$  no lo es.

Esta dicotomía nos conduce de forma natural a la caracterización de las asignaciones ortogonales realizables, la cual se realiza en términos de digrafos y esfericidad ascendente (*upward spherical*).

Sea  $H$  una asignación ortogonal de una inmersión cilíndrica  $P$  de un grafo 4-plano  $G = (V, A)$ , y sea  $r$  un segmento de  $H$  al cual se le ha asignado una dirección en la malla (horizontal o vertical) así como un sentido de

recorrido. Vamos a dotar de dirección y sentido al resto de los segmentos de  $H$  obteniendo así una *asignación ortogonal dirigida*, caracterizada por el hecho de que los caminos formados por segmentos con la misma dirección son caminos dirigidos. A partir de la dirección de  $r$  y teniendo en cuenta los campos  $a$  y  $s$  de la asignación dotamos de dirección en la malla al resto de los segmentos de  $H$ . El sentido de  $r$  hace dirigidos a los demás segmentos, teniendo en cuenta los sistemas de rotación en los vértices de la inmersión y siguiendo las siguientes pautas: la estructura deseada para las aristas incidentes en cada vértice es la mostrada en la Figura 2.6 (a) y los segmentos que definen un codo determinan distintos casos en función de su dirección y de la topología de la inmersión (ver Figura 2.6 (b)).

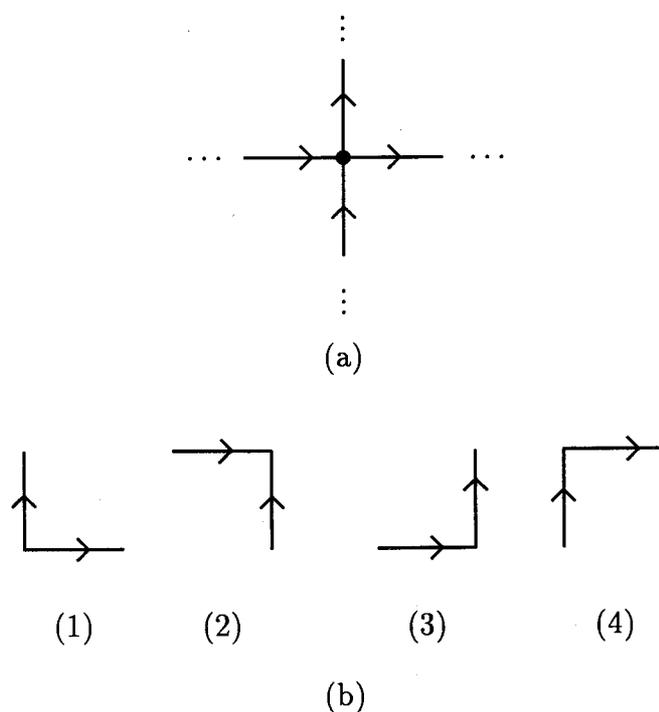


Figura 2.6: Estructuras formadas por los segmentos dirigidos de la asignación ortogonal.

Notemos que los casos (1) y (2) de los sentidos en los codos hacen que,

en general, las aristas de la asignación no sean dirigidas. En una asignación ortogonal dirigida son los segmentos los que están compuestos por vértice inicial y vértice final. Una característica importante de estas asignaciones es que los caminos compuestos por segmentos con la misma dirección son caminos dirigidos.

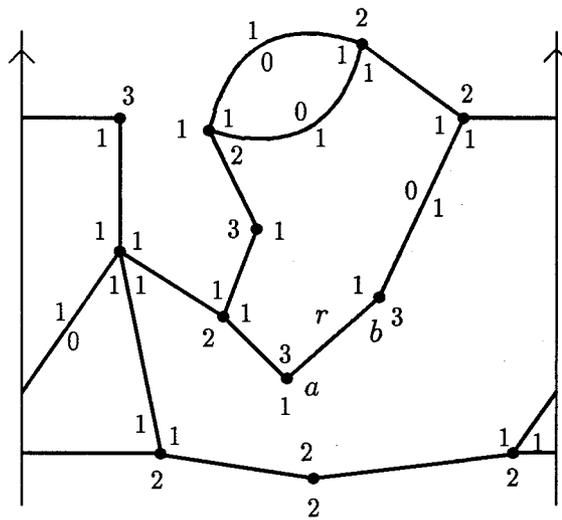


Figura 2.7: Inmersión cilíndrica dotada de una asignación ortogonal.

Se presentan los siguientes ejemplos para clarificar este proceso. Consideramos la asignación ortogonal mostrada en la Figura 1.7, la cual se explicita en la Figura 2.7 indicando los ángulos determinados por cada par de aristas incidentes y los codos que contiene. En ella se hace especial mención del segmento  $r$ , al cual se le dota de dos asignaciones de dirección y sentido que da lugar a las dos asignaciones ortogonales dirigidas de las Figuras 2.8 y 2.9.

Asociado a cada asignación ortogonal dirigida  $H$  se define el *digrafo*  $D_h = (V_h, A_h)$ , donde cada vértice de  $V_h$  es, o bien un vértice de  $V$  (si no es incidente a ningún segmento horizontal), o bien es el resultado de contraer en un punto cada camino maximal compuesto por segmentos horizontales de

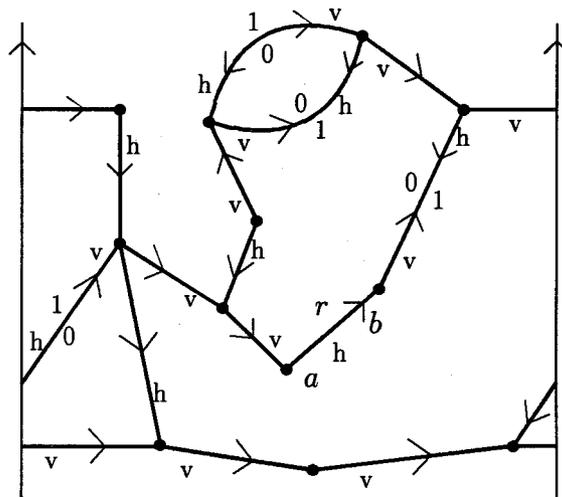


Figura 2.8: Asignación ortogonal dirigida obtenida a partir de la horizontalidad de  $r$  y sentido de  $a$  a  $b$ .

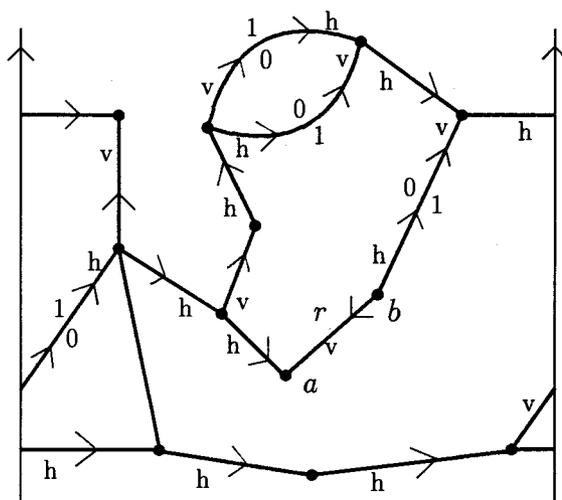


Figura 2.9: Asignación ortogonal dirigida obtenida a partir de la verticalidad de  $r$  y sentido de  $b$  a  $a$ .

$H$ . Recordemos que estos caminos son dirigidos según la construcción de  $H$ . El conjunto de aristas de  $D_h$  se obtiene eliminando de  $A$  los segmentos horizontales y añadiendo una arista dirigida por cada segmento vertical de  $H$  y en el mismo sentido. Los extremos de estas aristas serán, o bien vértices de  $H$  no incidentes en segmentos horizontales de la asignación dirigida, o bien vértices resultado de contracciones. Este último caso corresponde a la incidencia de vértices pertenecientes a caminos contraídos con otros vértices de la asignación.

Se llama *inmersión asociada a la asignación ortogonal dirigida  $H$* , y se denota  $\Gamma_h$ , a la inmersión de  $D_h$  que se obtiene a partir de  $P$  situando los vértices resultado de contracciones sobre el camino del que proceden y dibujando las nuevas aristas siguiendo la topología de la inmersión cilíndrica  $P$ . Las inmersiones  $\Gamma_h$  asociadas a las asignaciones ortogonales dirigidas de las Figuras 2.8 y 2.9 son las mostradas en las Figuras 2.10 y 2.11, respectivamente.

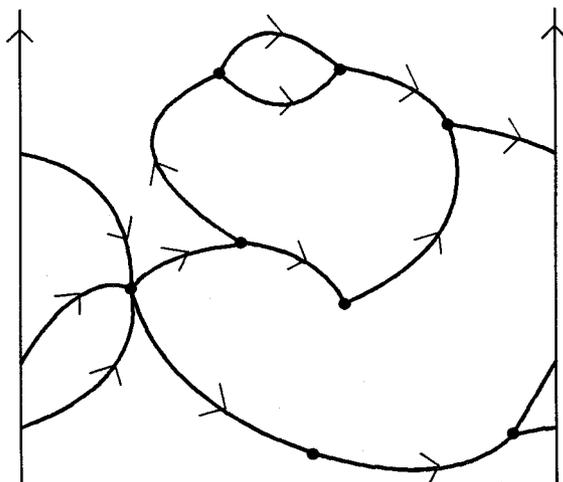


Figura 2.10: Inmersión asociada a la asignación ortogonal dirigida de la Figura 2.8.

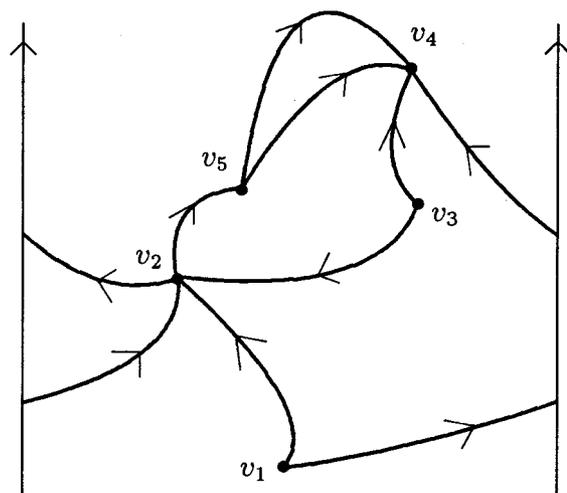


Figura 2.11: Inmersión asociada a la asignación ortogonal dirigida de la Figura 2.9.

Una inmersión cilíndrica  $\Gamma$  de un digrafo  $D$  decimos que es *esférica ascendente* si existe una inmersión equivalente sobre el cilindro de forma que todas sus aristas sean estrictamente ascendentes.

Bajo estos términos el teorema de caracterización de las asignaciones (flujos) ortogonales realizables se enuncia como sigue:

**Teorema 2.2.1.** *Una asignación ortogonal es realizable si y sólo si admite asignaciones de dirección y sentido de sus segmentos que la convierte en asignación dirigida cuya inmersión asociada es esférica ascendente.*

**Demostración:** En primer lugar vemos que si  $H$  es una asignación ortogonal realizable entonces se le puede asociar una inmersión esférica ascendente. Por ser  $H$  realizable está asociada a una inmersión en la malla  $P$ , luego sus segmentos llevan asignada intrínsecamente una dirección de horizontalidad o verticalidad. Consideramos los segmentos verticales como ascendentes y los horizontales recorridos de izquierda a derecha. Así,  $P$  pasa a ser un grafo dirigido de forma que la inmersión que resulta al contraer los caminos maximales formados por segmentos horizontales es esférica ascendente.

Para demostrar el recíproco partimos de una asignación ortogonal dirigida  $H$ , procedente de una inmersión cilíndrica  $P$ , de forma que la inmersión asociada  $\Gamma_h$  sea esférica ascendente. Esta propiedad hace que, aunque en general  $\Gamma$  es un multigrafo dirigido, no sea posible la existencia de lazos ni de ciclos dirigidos. Por cada conjunto de aristas múltiples entre dos vértices, mantengamos una de ellas subdividiendo las restantes al añadir vértices virtuales. El digrafo obtenido deja de ser un multigrafo y la inmersión asociada sigue siendo esférica ascendente.

Localizamos las fuentes y sumideros de  $\Gamma_h$ ; el vértice con menor altura será una fuente y el de mayor un sumidero. Consideramos una arista dirigida desde el Polo Sur hasta esta fuente y otra desde dicho sumidero al Polo Norte. Para las restantes fuentes hacemos llegar una arista dirigida hasta cada una

de ellas, procedente de un vértice con menor altura contenido en la misma cara que la fuente. De forma análoga, para eliminar los restantes sumideros, añadimos una arista dirigida desde cada uno de ellos hasta otro vértice de mayor altura contenido en la misma cara que el sumidero. Llamamos virtuales a estas aristas añadidas en las fuentes y sumideros.

Para simplificar la notación seguimos llamando  $D_h$  al digrafo que obtenemos después de las modificaciones, y  $\Gamma$  a la inmersión. Ésta sigue siendo esférica ascendente, y por tanto acíclica, y solamente contiene una fuente, el Polo Sur, y un sumidero, el Polo Norte, luego  $D_h$  es un st-grafo esférico.

La caracterización de los grafos que admiten una representación de visibilidad en la esfera, realizada por Tamassia y Tollis [69], asegura que el grafo resultante de eliminar las direcciones de las aristas de  $D_h$ , por su condición de st-grafo esférico, es bar-representable en la esfera y que es posible construir tal representación en tiempo lineal en el número de vértices de  $D_h$ , en definitiva de  $G$ . Esta representación de visibilidad en la esfera da lugar a una en el cilindro en la que los vértices de  $D_h$  son subconjuntos conexos de círculos máximos (en la representación cilíndrica plana un segmento horizontal), y las aristas son rayos de visibilidad a lo largo de generatrices (dirección vertical).

Llamamos intervalo de representación de visibilidad a los subconjuntos horizontales. A partir de esta representación de visibilidad vamos a obtener una inmersión en la malla asociada a  $H$ , equivalente a  $P$ . En primer lugar eliminamos los rayos de visibilidad correspondientes a las aristas virtuales. Seguidamente tenemos que solventar situaciones como las ilustradas en las Figuras 2.12 y 2.13.

Se realiza un barrido de Sur a Norte y de izquierda a derecha en los intervalos de la representación de visibilidad. Cada uno de estos intervalos, situados sobre círculos máximos, corresponde a un vértice de  $D_h$ , luego a un camino maximal de segmentos horizontales o a un vértice de  $P$  (original o virtual). Si estamos en este segundo caso (ver Figura 2.12), el intervalo correspondiente ve, a lo más, a otros dos intervalos, uno superior y otro inferior.

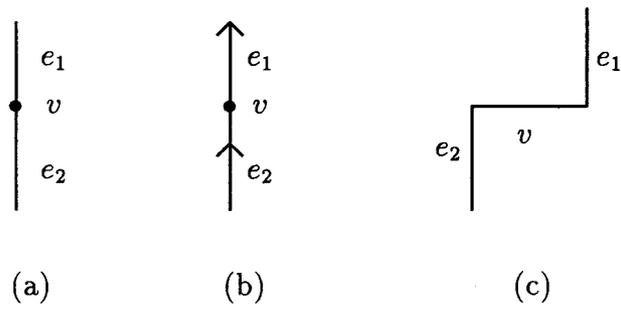


Figura 2.12: (a) Aristas incidentes en  $v$ ; (b) digrafo asociado; (c) posible estructura en la representación de visibilidad del digrafo.

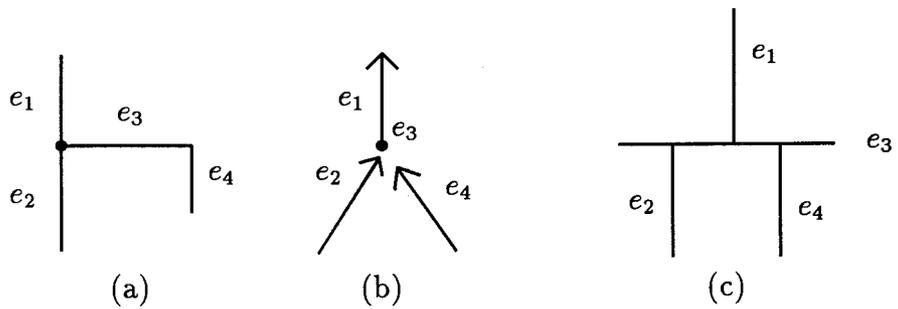


Figura 2.13: (a) Posibles incidencias en  $H$ ; (b) digrafo asociado; (c) posible estructura en la representación de visibilidad del digrafo.

Contraemos dicho intervalo en un punto, el de incidencia del rayo de visibilidad inferior. Notemos que el rayo superior quedaría desplazado, situación que queda resuelta en el siguiente paso.

Si el intervalo corresponde a un camino horizontal (ver Figura 2.13) tenemos que trasladar rayos de visibilidad para que incidan sobre el mismo punto del intervalo. Notemos que el orden en el que están situados los rayos superiores o inferiores no hay que alterarlo puesto que  $\Gamma_h$  y la representación de visibilidad poseen la misma estructura topológica, y por definición  $\Gamma_h$  conserva la topología de la inmersión cilíndrica asociada a  $H$ . Los rayos inferiores se han considerado en una etapa anterior, luego en este punto no se modifican. Realizamos un barrido, de izquierda a derecha, de los rayos de visibilidad superiores del intervalo  $I$  correspondiente. De las adyacencias en  $P$  sabemos si un rayo tiene que situarse sobre la misma generatriz que alguno inferior, o si por el contrario está situado en un extremo del intervalo o entre dos rayos inferiores. En cualquiera de los casos sea  $R$  el rayo de visibilidad que tenemos que trasladar. Consideramos la semibanda  $S$  delimitada por  $I$ , la generatriz  $g$  que contiene a  $R$  y la que contiene al rayo superior incidente en  $I$ , inmediatamente a la derecha de  $R$  o a la izquierda, según sea el desplazamiento a la derecha o izquierda respectivamente. Situamos  $R$  en su posición correspondiente contrayendo la semibanda  $S$  en dirección horizontal y prolongando la parte izquierda (derecha) de los intervalos intersectados por  $g$  (ver Figura 2.14).

Los vértices de la inmersión buscada vienen dados por puntos de los intervalos: los de incidencia de los rayos de visibilidad y posiblemente otros puntos correspondientes a vértices que no son incidentes en  $H$  a segmentos verticales. Estos datos se obtienen de la inmersión cilíndrica  $P$ . Los rayos de visibilidad determinan los segmentos verticales y los horizontales quedan constituidos por fragmentos de los intervalos de visibilidad.

Hemos obtenido una inmersión equivalente a  $P$  en la malla del cilindro, cuyos ángulos y codos se corresponden con los de la asignación ortogonal  $H$ ,

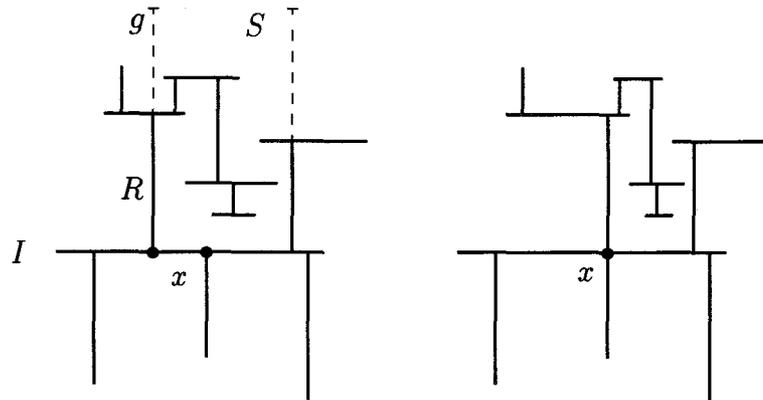


Figura 2.14: Situamos los rayos de visibilidad según las adyacencias en el vértice  $x$ .

luego ésta es realizable. ■

Este teorema, además de caracterizar las asignaciones ortogonales realizables, proporciona un algoritmo lineal que construye una inmersión ortogonal equivalente a la dada a partir de la inmersión  $\Gamma_h$  asociada a la asignación ortogonal dirigida y realizable.

La existencia de ciclos dirigidos contenidos en la inmersión dirigida de la Figura 2.10 hace que ésta no sea esférica ascendente, luego la asignación ortogonal dirigida de la que procede no es realizable. Por el contrario, la de la Figura 2.11 sí lo es, tal y como muestra el gráfico 2.15. A partir de esta inmersión esférica ascendente se ha obtenido en la Figura 2.16 una inmersión ortogonal de la asignación ortogonal dada en la Figura 2.7.

Notemos que si al segmento  $r$  le asignamos el sentido contrario al dado, la inmersión dirigida asociada sigue siendo esférica ascendente si intercambiamos los papeles de las caras externas, luego con dicho sentido la asignación ortogonal dirigida también es realizable y la inmersión ortogonal que origina

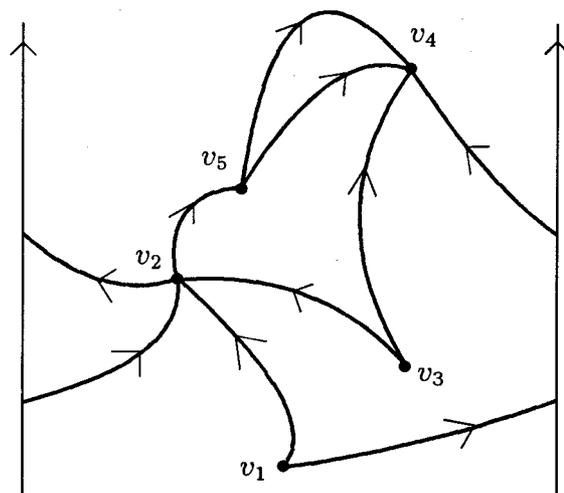


Figura 2.15: Dibujo esférico ascendente de la inmersión de la Figura 2.11.

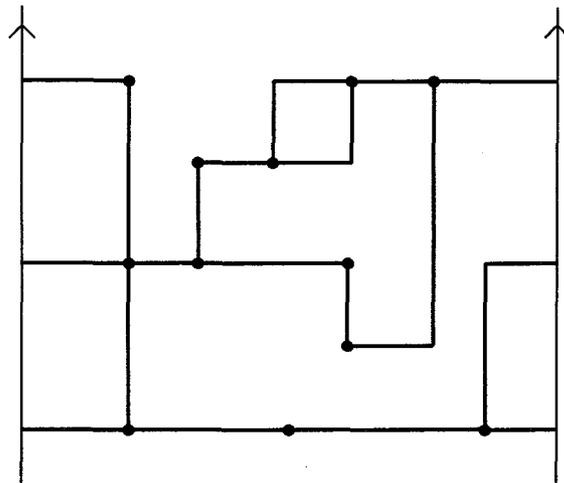


Figura 2.16: Inmersión ortogonal de la asignación ortogonal dirigida y realizable de la Figura 2.9.

es igual a la dada cambiando los polos que contienen las caras externas. Como conclusión, si una asignación ortogonal dirigida es realizable, también lo es si cambiamos el sentido de los segmentos.

Una vez que sabemos identificar las asignaciones ortogonales realizables, siendo éstas las que nos interesan, pues nos conducen a inmersiones ortogonales, vamos a caracterizar teóricamente, también en términos de flujos, la óptima, aquella que contiene el menor número de codos entre todas las asociadas a la misma inmersión cilíndrica  $P$ . La idea general es comenzar con una inmersión ortogonal equivalente a  $P$ , con un número de codos arbitrario, y después ir disminuyendo tal número. En esta línea el siguiente resultado proporciona una inmersión ortogonal de un grafo 4-plano equivalente a una dada, con  $4n$  codos, siendo  $n$  el número de vértices del grafo. Su demostración conduce a un algoritmo que corre en tiempo  $O(n)$ .

**Lema 2.2.2.** *Dada una inmersión cilíndrica  $P$  de un grafo 4-plano de  $n$  vé-*

*tices, es posible obtener en tiempo  $O(n)$ , una inmersión ortogonal equivalente a  $P$  con a lo más  $4n$  codos.*

**Demostración:** Podemos suponer que el grafo  $G$  es conexo, tal y como indicamos anteriormente. Queremos obtener una representación bar-cilíndrica de  $G$ , es decir una representación de visibilidad en el cilindro, equivalente a  $P$ . En tiempo lineal [71] en el número de vértices se puede saber si un grafo es bar-cilíndrico, y en caso afirmativo obtener una representación de bar-visibilidad en el cilindro equivalente a la inmersión dada. En cualquier caso se asegura que todo grafo 2-conexo es bar-cilíndrico. En virtud de estos resultados comprobamos si  $G$  es bar-cilíndrico, y si lo es obtenemos una representación de bar-visibilidad en el cilindro equivalente a  $P$ .

En caso contrario se define un grafo 2-conexo  $G'$  a partir de  $G$ , añadiendo aristas, del cual podremos obtener una inmersión bar-cilíndrica. Localizamos los vértices de corte de  $G$  y consideramos sus sistemas de rotación. En cada vértice de corte recorreremos las aristas incidentes en el sentido de la rotación y conectamos los extremos de dos aristas consecutivas si éstos pertenecen a bloques distintos, aún no considerados en el recorrido. De esta forma obtenemos una inmersión cilíndrica  $P'$  de un grafo  $G'$  2-conexo en tiempo lineal.

El algoritmo VISIB-2C [71], aplicado a  $P'$ , nos proporciona una representación bar-cilíndrica,  $K$ , de  $G'$ , equivalente a  $P'$ . En ella los vértices han sido representados mediante segmentos sobre generatrices (intervalos de visibilidad) y los rayos de visibilidad son horizontales. Notemos que todos los intervalos de visibilidad de  $K$  se corresponden con vértices de  $G$ , sin embargo, al haber añadido aristas a  $G$  para construir  $G'$ , aparecerán en  $K$  vértices visibles que no constituyen una arista de  $G$ . En este sentido decimos que  $K$  es una representación de visibilidad débil de  $P$  en el cilindro.

Como  $G$  es 4-plano cada intervalo  $u$  de  $K$  es débilmente visible, a lo más, por cuatro intervalos de la representación, cuyos rayos de visibilidad definen, al

incidir, cuatro puntos distintos de  $u$ . Situamos el vértice de  $G$  correspondiente al intervalo en uno de los puntos intermedios, o en cualquier extremo en el caso de la incidencia de dos únicos rayos. Las distintas posibilidades que se pueden tener, dependiendo de la dirección de los rayos de visibilidad, son las mostradas en la Figura 2.17. Los restantes casos se obtienen por simetría de alguna de las seis configuraciones consideradas.

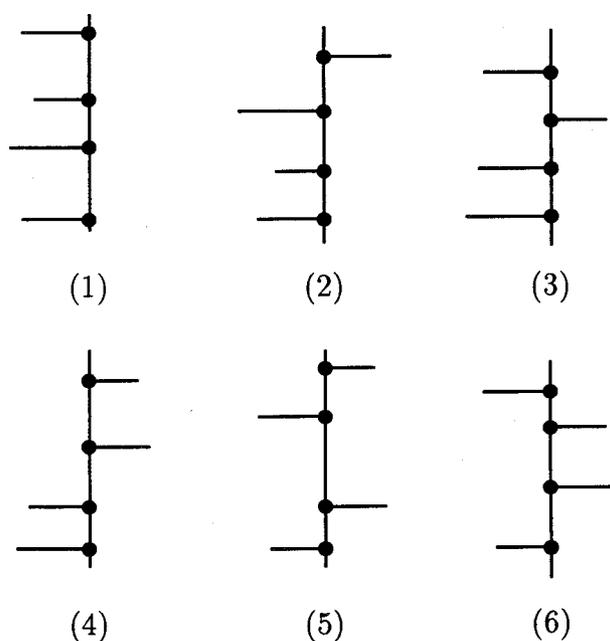


Figura 2.17: Distintas configuraciones de visibilidad.

A partir de  $K$  construimos una inmersión ortogonal  $Q$ , equivalente a  $P$ . Los vértices de  $Q$  ya los hemos fijado y las aristas se definen como se muestra en la Figura 2.18, dependiendo de la configuración de visibilidad de cada intervalo.

Notemos que por cada vértice hemos obtenido cuatro codos, luego  $Q$  contiene en total  $4n$  codos. Cada arista presenta como máximo cuatro codos.

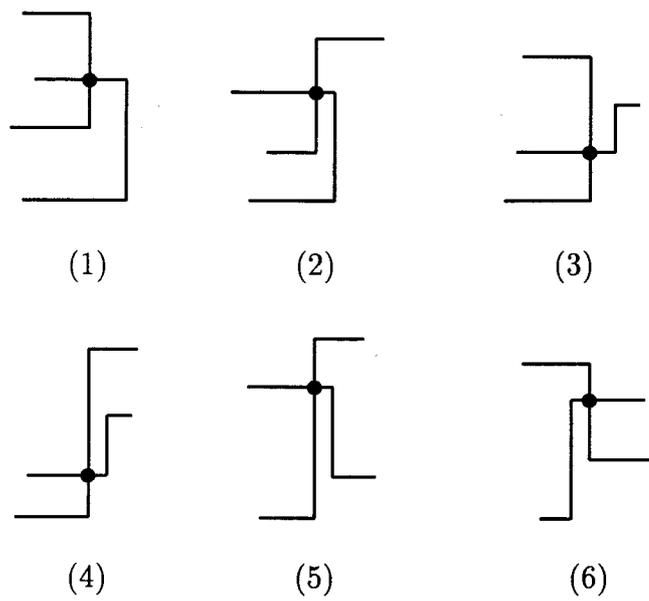


Figura 2.18: Inmersión ortogonal a partir de la configuración de visibilidad.

Todos los pasos realizados en la demostración requieren tiempo lineal, luego la construcción de una asignación ortogonal realizable a partir de una inmersión cilíndrica de un grafo 4-plano se realiza en tiempo lineal. ■

Conservando la inmersión, hemos obtenido una representación de  $P$  en la malla, es decir, hemos dibujado el grafo de forma que los segmentos de las aristas están contenidos en círculos máximos o generatrices del cilindro. Esta inmersión en la malla determina una asignación ortogonal realizable, pero además, fija la dirección de los segmentos de las aristas.

Para un segmento de la asignación ortogonal, existen dos posibilidades en cuanto a su dirección, vertical u horizontal. Obsérvese, además, que fijando la dirección de uno de los segmentos, a partir de la asignación ortogonal, es posible determinar la dirección del resto de los segmentos. En este sentido diremos que una asignación ortogonal realizable  $H$  es *bidireccional* si existen dos inmersiones ortogonales equivalentes cuyos ángulos y codos coinciden con  $H$  y de forma que la diferencia entre ellas radica en la dirección de sus segmentos. En caso contrario diremos que es *unidireccional*.

La bidireccionalidad o unidireccionalidad de una asignación radica en la existencia de ciclos esenciales en la inmersión, los cuales determinan el carácter cilíndrico de ésta. El estudio de las asignaciones ortogonales según este carácter nos permitirá caracterizar la óptima de forma más simple.

Sea  $C$  un ciclo y  $P$  una inmersión ortogonal y esencial de él. Fijamos un sentido de recorrido  $\tau$  de los círculos máximos (paralelos) del cilindro y recorremos  $P$  en el sentido que nos indica  $\tau$ . Diremos que un segmento horizontal de  $P$  es *positivo* si se recorre en el mismo sentido que  $\tau$ , en caso contrario diremos que es *negativo*. Notemos que el carácter de los segmentos horizontales será el mismo para los dos sentidos de recorrido posibles.

Un segmento vertical es *ascendente* si se recorre en dirección norte, y *descendente* en dirección sur.

En este contexto enunciamos y demostramos el siguiente lema:

**Lema 2.2.3.** *Toda inmersión ortogonal y esencial de un ciclo cuyos segmentos horizontales sean positivos da lugar a una asignación ortogonal realizable y unidireccional.*

**Demostración:** Si cambiamos la dirección de las aristas de la inmersión los segmentos horizontales pasan a ser segmentos verticales, y para obtener la misma asignación ortogonal en la malla, como partimos de segmentos horizontales positivos, todos los verticales serán, o bien ascendentes o bien descendentes. Si todos los segmentos verticales tienen el mismo carácter, la inmersión obtenida no es factible, no es posible obtener un ciclo, sino un camino cuyos extremos son distintos. Por tanto no podemos obtener otra inmersión del ciclo en la malla con la misma asignación ortogonal, intercambiando horizontalidad y verticalidad. ■

Realizando un estudio en la misma línea para inmersiones cualesquiera obtenemos lo siguiente.

**Teorema 2.2.4.** *Una asignación ortogonal asociada a una inmersión ortogonal cilíndrica de un grafo 4-plano es bidireccional si y sólo si la inmersión es plana.*

**Demostración:** La condición suficiente se sigue del estudio realizado en el plano [64].

Para la condición necesaria tenemos que probar que ninguna inmersión en las condiciones del teorema contiene ciclos esenciales. Por reducción al absurdo, supongamos que existe una inmersión ortogonal cilíndrica  $P$  que da lugar a una asignación bidireccional y que contiene un ciclo esencial. Partimos, por tanto, de un ciclo  $C$  y dos dibujos  $P_1$  y  $P_2$  de él en la malla con la misma

asignación ortogonal, cuya diferencia es la dirección de los segmentos de las aristas. Sin pérdida de generalidad podemos situar vértices ficticios en cada uno de los codos de la representación de  $C$ .

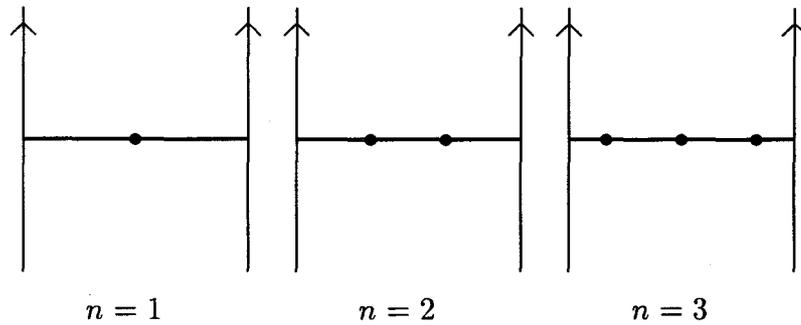


Figura 2.19: Ciclos esenciales con menos de tres vértices.

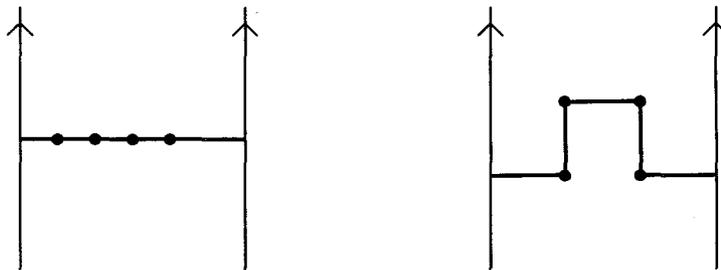


Figura 2.20: Ciclos esenciales con cuatro vértices.

Si  $P_1$  contiene una arista vertical, presenta al menos cuatro vértices, puesto que las inmersiones ortogonales esenciales de ciclos con tres o menos vértices son las representadas en la Figura 2.19, en las que todas las aristas son horizontales. En estos ciclos, al cambiar la dirección de las aristas pasan a ser verticales por lo que no pueden constituir ningún ciclo (Lema 2.2.3).

Si  $P_1$  contiene cuatro vértices existen dos posibilidades; por una parte

un ciclo con todas las aristas horizontales, y si existe alguna vertical, el ciclo esencial posible, salvo simetrías, es el mostrado en la Figura 2.20.

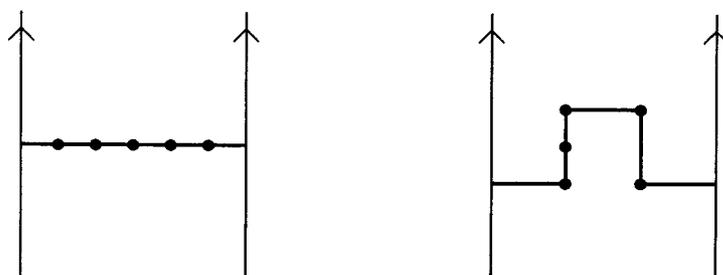


Figura 2.21 : Ciclos esenciales con cinco vértices.

A partir de estos dos ciclos no es posible obtener otros cambiando la dirección de las aristas; para el primero porque obtendríamos todas las aristas verticales, y aplicando el Lema 2.2.3 obtenemos el resultado para el segundo. Los ciclos esenciales de cinco vértices se obtienen realizando la subdivisión de una arista de un ciclo de cuatro vértices, por lo tanto no admiten inmersiones equivalentes con la misma asignación ortogonal y distinta dirección de las aristas (ver Figura 2.21).

Se pueden obtener ciclos de seis vértices en la malla por subdivisión de una arista de un ciclo de cinco vértices, pero no son éstos los únicos existentes, tal y como muestra la Figura 2.22. En ella se presentan los restantes casos de ciclos de seis vértices, salvo simetrías. El ciclo mostrado en la Figura 2.22(a) se encuentra en las condiciones del Lema 2.2.3, por lo que no admite un cambio de dirección en las aristas. Para el otro ciclo razonamos de la siguiente forma. Notemos que la arista  $e = (a_2, a_3)$  es negativa para cualquier sentido de recorrido, en particular para el considerado. Vamos a definir una inmersión ortogonal  $Q_1$  de un ciclo esencial  $C'$ , a partir de  $P_1$ , de forma que la existencia de  $P_2$  nos determine una inmersión  $Q_2$  de  $C'$  con la misma asignación ortogonal que  $Q_1$ , cambiando la dirección de las aristas. Prolongamos la semirrecta

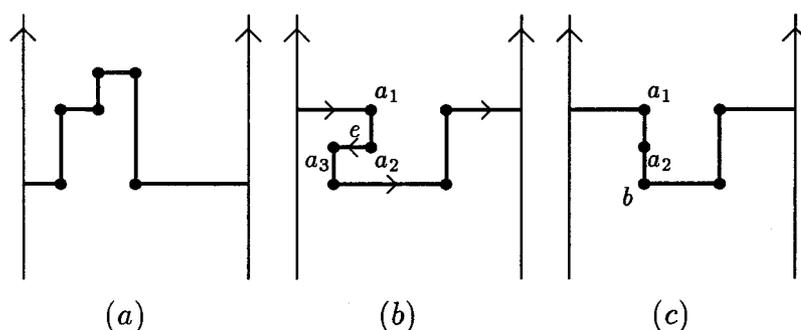


Figura 2.22: Dos tipos de ciclos esenciales de seis vértices y reducción de uno de ellos.

vertical de origen  $a_1$  que contiene a  $a_2$ , hasta que intersecte a otra arista de  $P_1$ , bien sea en un punto intermedio o en un extremo, determinando así un punto  $b$  de  $P_1$ . La existencia de  $b$  está garantizada por el hecho de que  $P_1$  es un ciclo esencial, luego tiene que recorrer todo el cilindro.  $Q_1$  queda definido al eliminar el camino que va de  $a_2$  a  $b$  en  $C$  y sustituirlo por la arista vertical  $(a_2, b)$  (ver Figura 2.22(c)).

Al camino que se elimina, correspondiente a la arista negativa de  $P_1$ , se le llama *bolsillo de extremos*  $a_2$  y  $b$ . Realizando el mismo proceso en  $P_2$ , cambiando la noción de verticalidad por horizontalidad, obtenemos una inmersión ortogonal  $Q_2$  de  $C'$  equivalente a  $Q_1$ , las cuales difieren en la dirección de sus aristas. La inmersión  $Q_1$  de  $C'$  se encuentra en las condiciones del Lema 2.2.3, luego la existencia de  $Q_2$ , que se traduce a la de  $P_2$ , nos lleva a una contradicción.

Supongamos que el ciclo  $C$  tiene  $n$  vértices. Si  $P_1$  no contiene aristas negativas, el Lema 2.2.3 contradice la existencia de  $P_2$ . Supongamos que al recorrer  $P_1$  aparecen aristas negativas. Eliminamos el bolsillo de una de ellas, y lo sustituimos por una arista vertical, tal y como vimos anteriormente. Notemos que de esta forma se pueden suprimir varias aristas negativas, todas las

pertenecientes al bolsillo de la considerada. En cualquier caso, el número de vértices disminuye al realizar este proceso. Así se define una inmersión ortogonal  $Q_1$ , de un ciclo  $C'$  con un número de vértices menor que  $n$ . Eliminamos de  $P_2$  las aristas correspondientes al bolsillo suprimido en  $P_1$ , sustituyéndolas por una arista horizontal de vértices los extremos del bolsillo. Queda así definida una inmersión ortogonal  $Q_2$  en la malla, del ciclo  $C'$ , equivalente a  $Q_1$ . Las aristas comunes a  $P_1$  y  $Q_1$  también lo son en  $P_2$  y  $Q_2$ , pero con distinta dirección, y la arista vertical añadida en  $Q_1$  pasa a ser horizontal en  $Q_2$ . Obtenemos por tanto, que la diferencia entre  $Q_1$  y  $Q_2$  es la dirección de las aristas. La existencia de  $Q_2$ , y por tanto la de  $P_2$ , no es factible al aplicar la hipótesis de inducción a  $Q_1$ .

Concluimos que una inmersión ortogonal de un grafo 4-plano en las condiciones dadas no puede contener un ciclo esencial. ■

Puesto que nuestro problema se centra en inmersiones cilíndricas no planas, en virtud de los resultados anteriores, la existencia de ciclos esenciales hará que las asignaciones ortogonales realizables que consideremos sean unidireccionales. Por tanto, dada una asignación ortogonal, ésta puede no ser realizable para ambas direcciones sobre un segmento, realizable para una y para otra no serlo, pero en ningún caso, si trabajamos con inmersiones cilíndricas propiamente dichas, puede ser realizable para las dos direcciones.

Retomando los resultados considerados sobre Teoría de flujos en redes, recordamos la equivalencia entre asignación ortogonal de una inmersión cilíndrica  $P$  y flujo en la red  $N(P)$  de valor  $z(P)$ . En esta relación el número de codos de la asignación coincide con el coste del flujo correspondiente, luego obtener la asignación óptima es equivalente a obtener el flujo de mínimo coste. El Lema 1.2.5 caracteriza dicho flujo como aquel que no admite ciclos aumentadores de coste negativo. El Teorema 1.2.6 presenta el concepto equivalente a ciclo aumentador de coste negativo para asignaciones ortogonales, este es, curva sobre el cilindro bajo ciertas condiciones relativas a la asignación, y por tanto caracteriza la asignación ortogonal óptima.

Puesto que vamos encaminados a caracterizar la asignación ortogonal realizable que proporciona el mínimo número de codos, tenemos que estudiar como varían estos resultados sobre flujos en redes cuando particularizamos el estudio a dichas asignaciones.

Hablamos de *flujos ortogonales realizables* cuando nos referimos al flujo que proporciona el Lema 1.2.1 a partir de una asignación ortogonal realizable.

En general, la aplicación de un ciclo aumentador de coste negativo a un flujo ortogonal realizable no tiene que originar necesariamente otro flujo realizable. Bajo esta perspectiva decimos que un ciclo aumentador de coste negativo en un flujo ortogonal se llama *realizable* si el flujo ortogonal a que da lugar es realizable.

La caracterización de los flujos ortogonales realizables óptimos viene dada por el siguiente teorema.

**Teorema 2.2.5.** *Un flujo ortogonal realizable asociado a una inmersión cilíndrica  $P$  es de mínimo coste entre todos los flujos ortogonales realizables asociados a  $P$  si y sólo si no contiene ciclos aumentadores del flujo de coste negativo, esenciales y realizables.*

**Demostración:** El Lema 1.2.5 garantiza la condición necesaria de forma inmediata, al ser los ciclos considerados de coste negativo y realizables.

Para demostrar la condición suficiente vamos a probar que dado un ciclo aumentador del flujo de coste negativo, asociado a un flujo ortogonal  $x$  existe otro ciclo aumentador asociado a  $x$  de coste negativo, esencial y realizable, de forma que ambos ciclos aplicados a  $x$  originan el mismo flujo. Con este resultado y en virtud del Lema 1.2.5 se obtiene la condición a probar.

Sea  $C$  un ciclo aumentador de coste negativo asociado a  $x$ . Si  $C$  no es realizable el flujo  $x'$  obtenido a partir de  $x$  tiene menor coste pero no es realizable. Nuestro objetivo es caracterizar el flujo ortogonal realizable de

mínimo coste, sin embargo, aunque  $x'$  no sea realizable, hay que considerarlo, ya que puede admitir un ciclo  $C'$  de coste negativo que de lugar a otro flujo  $x''$  realizable. En esta situación demostramos que a partir de  $C$  y  $C'$  es posible definir un ciclo aumentador  $C''$ , asociado a  $x$ , de coste negativo que da lugar al flujo  $x''$ , luego un ciclo realizable. Tanto  $C$  como  $C'$  son ciclos de la red  $N(P)$  luego la relación entre ellos vendrá dada por los siguientes casos:

Caso 1)  $C$  y  $C'$  comparten vértices pero no aristas,

Caso 2)  $C$  y  $C'$  comparten aristas,

Caso 3)  $C$  y  $C'$  son disjuntos.

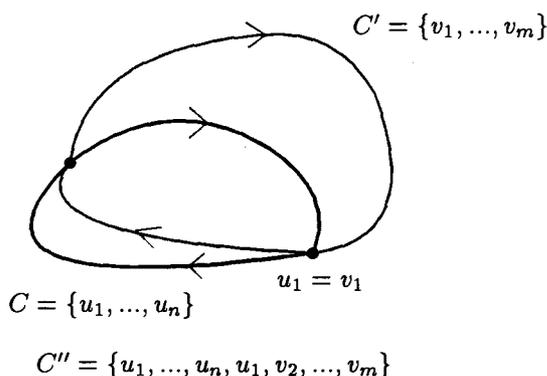


Figura 2.23: Ciclos aumentadores de coste negativo que solamente comparten vértices.

En el primer caso  $C''$  se define uniendo los dos ciclos por uno de los vértices comunes y teniendo en cuenta el sentido de recorrido de cada uno (ver Figura 2.23). El hecho de no compartir aristas hace que  $C''$  sea de coste negativo para  $x$  y que de lugar a  $x''$ .

En el caso 2 tenemos en cuenta si las aristas comunes se recorren en el mismo sentido en  $C$  y  $C'$ , es decir si tienen el mismo carácter (directas

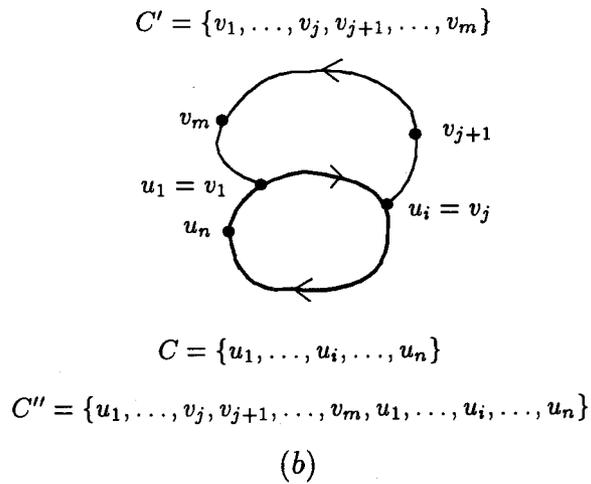
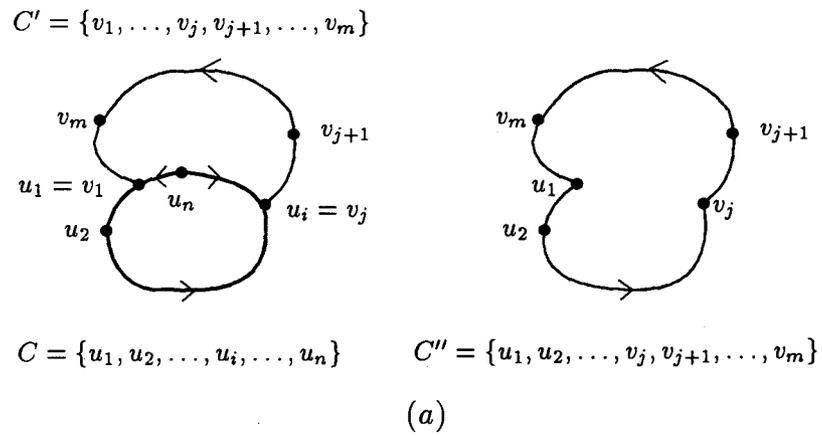


Figura 2.24: Ciclos aumentadores de coste negativo con aristas comunes: (a) aristas con carácter distinto en cada ciclo; (b) aristas con el mismo carácter en ambos ciclos.

o inversas) para  $x$  y  $x'$  respectivamente. El recorrido en distinto sentido se traduce en que los flujos  $x$  y  $x''$  toman los mismos valores en dichas aristas, la unidad de flujo que aumenta un ciclo la disminuye el otro. En esta situación se define  $C''$  eliminando de  $C \cup C'$  las aristas comunes, como nos ilustra la Figura 2.24 (a). Cada arista eliminada tiene distinto carácter en  $C$  y  $C'$ , luego el número de aristas directas eliminadas es el mismo que inversas, por lo que  $C''$  sigue siendo un ciclo aumentador de coste negativo para el flujo  $x$ . La aplicación de  $C''$  a  $x$  da lugar a  $x''$  luego es un ciclo aumentador de coste negativo realizable. En el caso en el que el recorrido sea el mismo realizamos la unión de los dos ciclos para definir  $C''$ , recorriendo así dos veces las aristas comunes (ver Figura 2.24 (b)). Así definido  $C''$  es un ciclo aumentador del flujo de coste negativo para  $x$ . Esta composición de ciclos da lugar a  $x''$  luego  $C''$  es realizable.

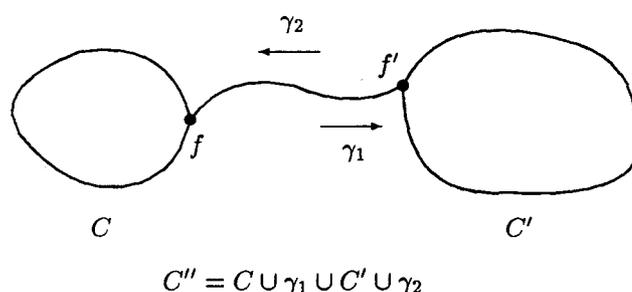


Figura 2.25: Ciclos aumentadores de coste negativo disjuntos.

Nos queda estudiar el caso en el que los dos ciclos sean disjuntos. Vamos a unir  $C$  y  $C'$  por un camino en  $N(P)$  para obtener un nuevo ciclo de forma que aplicado a  $x$  nos de  $x''$ . En la red no existen vértices, que sean también de  $P$  y adyacentes entre sí, luego todo ciclo en  $N(P)$  contiene al menos un vértice que es una cara de  $P$ . Elegimos  $f$  y  $f'$ , dos vértices de  $C$  y  $C'$  respectivamente, correspondientes a caras de  $P$ . La conexión de la inmersión  $P$  considerada nos permite elegir un camino en  $N(P)$  entre  $f$  y  $f'$ , que denotamos  $\gamma_1$ , no dirigido y formado por aristas de  $A_F$ . Como dos caras de  $P$  son adyacentes en la red

mediante dos aristas dirigidas en distinto sentido, siempre es posible considerar como directas las aristas de  $\gamma_1$ . Llamamos  $\gamma_2$  al camino  $\gamma_1$  recorrido de  $f'$  a  $f$ , por lo que estará formado por aristas inversas. Definimos  $C'' = C \cup \gamma_1 \cup C' \cup \gamma_2$  (ver Figura 2.25). Este ciclo aumentador es de coste negativo para  $x$  puesto que  $C$  y  $C'$  lo eran y los caminos añadidos aportan el mismo número de aristas directas que inversas. La unidad de flujo que aumenta una arista de  $\gamma_1$  queda disminuida por la correspondiente arista inversa de  $\gamma_2$ , luego  $C''$  aplicado a  $x$  da lugar a  $x''$ . Hemos definido por tanto, un ciclo aumentador de coste negativo y realizable.

Por último nos queda analizar el caso en el que  $C$  sea homotópicamente no nulo y realizable, obteniendo un nuevo ciclo aumentador  $C^*$  de coste negativo homotópicamente nulo y realizable. El análisis anterior nos garantiza que, sin pérdida de generalidad, podemos suponer que  $C$  está definido en un flujo ortogonal realizable  $x$ , es decir, flujo asociado a una inmersión en la malla, y por ser realizable sabemos que su aplicación dará lugar a otra inmersión ortogonal  $x'$ . El ciclo  $C$ , por ser esencial, divide a la asignación ortogonal asociada a  $x$  en dos partes conteniendo un polo cada una. La aplicación del ciclo sobre  $x$  hará que se modifiquen los campos  $a$  y  $s$  de las aristas atravesadas por  $C$ , luego en una de dichas partes cambia la asignación de horizontal y vertical de sus segmentos, al pasar a  $x'$ . Si llamamos  $A$  a la parte afectada, la asignación ortogonal de los segmentos de  $A$  no cambia de  $x$  a  $x'$  (excepto la de los atravesados por  $C$ ), pero sí la dirección, por lo que tenemos una inmersión en la malla bidireccional. En virtud del Lema 2.2.4,  $A$  no contiene ciclos esenciales de  $P$ , es decir existe una cara en  $A$  atravesada por  $C$  que contiene círculos máximos del cilindro, y como partimos de un grafo conexo podemos suponer que dicha cara es exterior. Definimos la curva homotópicamente nula  $C^*$  sustituyendo el arco de  $C$  contenido en la cara exterior de  $A$  por un arco que rodee a la inmersión tal y como nos indica la Figura 2.26.  $C$  y  $C^*$  tienen el mismo coste y dan lugar al mismo flujo ya que atraviesan las mismas aristas de la inmersión, luego  $C^*$  es un ciclo aumentador de coste negativo homotópicamente nulo y realizable asociado a  $x$ .

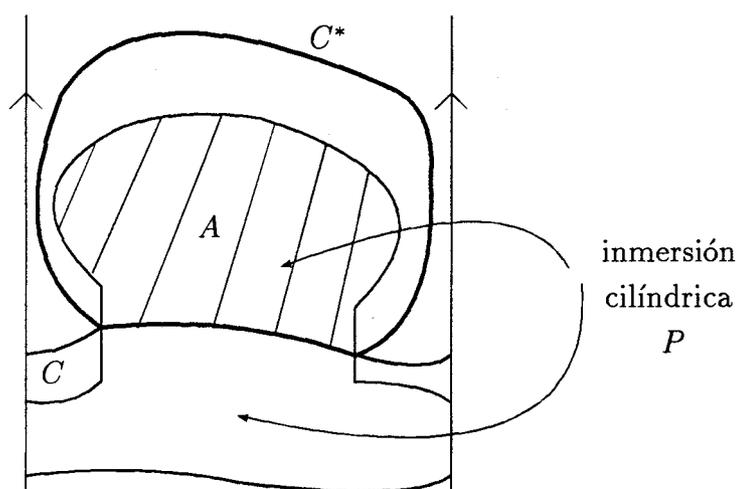


Figura 2.26: Los ciclos  $C$  y  $C^*$  dan lugar al mismo flujo, siendo  $C^*$  homotópicamente nulo.

Como conclusión, para obtener un flujo ortogonal realizable óptimo solamente hay que tener en cuenta los ciclos aumentadores del flujo que sean realizables y no esenciales. ■

El Teorema 1.2.6 caracterizaba la asignación ortogonal óptima, asociando a cada ciclo aumentador de coste negativo sobre un flujo, una curva con ciertas características sobre la asignación ortogonal. En la misma línea, el siguiente resultado asocia una curva particular a cada ciclo aumentador del flujo realizable y no esencial, caracterizando así, a la asignación ortogonal realizable óptima.

**Teorema 2.2.6.** *Una asignación ortogonal realizable  $H$  asociada a una inmersión cilíndrica  $P$  de un grafo 4-plano  $G$  tiene el mínimo número de codos entre todas las asignaciones ortogonales realizables asociadas a  $P$  si y sólo si no existe una curva cerrada dirigida  $J$ , en general no simple, tal que*

- a)  $J$  intersecta a cada arista de  $P$  a lo más cuatro veces.
- b)  $J$  entra a vértices y codos de  $P$  cuyos campos  $a$  y  $s$  son mayores o iguales a 2, o iguales a 1, respectivamente.
- c) Más de la mitad de los cruces entre  $J$  y las aristas de  $P$  se producen en aristas que forman un codo de asignación 1, ángulo mayor, en la cara por la que entra la curva; estos cruces tienen lugar en dicho codo.

Además, si existe una curva  $J$  con estas características, se puede encontrar una asignación ortogonal realizable  $H'$  asociada a  $P$  tal que

- a)  $H'$  tiene menos codos que  $H$ ;
- b)  $H$  y  $H'$  solamente se diferencian en las aristas y vértices cruzados por  $J$ .

**Demostración:** Puesto que la prueba sigue los mismos pasos generales que la del Teorema 1.2.6, solamente señalaremos las diferencias existentes entre ambas. La construcción de la curva  $J$ , a partir de un ciclo aumentador realizable y de coste negativo, se realiza de forma análoga, teniendo en cuenta que ésta puede cortar a una misma arista de  $P$  a lo más cuatro veces, e imponiendo que los arcos de curva sobre el cilindro correspondientes a la misma arista de la inmersión sean distintos, para no perder información en cuanto al ciclo aumentador.

La segunda diferencia esencial es la existencia de posibles autointersecciones en la curva, bien en vértices de  $P$  o bien en puntos asociados a caras, correspondientes a vértices repetidos en el ciclo sin repetición de aristas.

Puesto que la construcción de la curva definida en el Teorema 1.2.6 seguía la topología de la inmersión  $P$ , tenemos información sobre si el ciclo es esencial o no. ■

Para comprobar si una asignación ortogonal realizable es óptima, en virtud de los dos resultados anteriores, solamente tenemos que considerar las curvas no esenciales; y si la aplicación de tal curva conduce a un flujo no realizable, no tenemos que tenerla en cuenta y tomar solamente las realizables.

El proceso para obtener una asignación ortogonal realizable óptima, es decir una inmersión ortogonal equivalente a  $P$  con el mínimo número de codos, consistiría en, a partir de la inmersión ortogonal equivalente a  $P$  que nos proporciona el Lema 2.2.2, localizar un ciclo aumentador del flujo de coste negativo realizable y no esencial, cuya aplicación al flujo actual origina una nueva asignación ortogonal realizable. Este proceso se repetiría hasta llegar a una asignación ortogonal realizable que no admita tales ciclos, la cual sería óptima según el Teorema 2.2.5.

El paso de aplicar un ciclo  $C$  a un flujo  $x$  consiste en disminuir una unidad el ángulo por donde entra, aumentándolo por donde sale. Para comprobar si el nuevo flujo  $x'$  es realizable o no (Teorema 2.2.1), tenemos que asignar dire-

cciones a sus segmentos. Puesto que  $x$  es realizable, conocemos las direcciones que presentan sus segmentos en una inmersión ortogonal suya, luego el aplicar  $C$  a  $x$  va a suponer determinar cual de los dos segmentos que constituyen un ángulo, cambia de dirección. Las dos posibilidades no son factibles ya que conducirían a dos inmersiones ortogonales con la misma asignación de ángulos y codos, pero con distintas direcciones en sus segmentos, y puesto que  $P$  contiene algún ciclo esencial, el Lema 2.2.4 no hace posible esta situación. Por tanto, dado un ciclo  $C$ , para comprobar si  $x'$  es realizable hay que considerar dos digrafos asociados, obtenidos de los dos posibles cambios de los segmentos que forman un ángulo por el que entra  $C$ , y comprobar si alguno (a lo más uno) conduce a una asignación ortogonal dirigida realizable. Esta dicotomía se traduce en considerar el cambio de dirección de los segmentos que cruza  $C$  y que se encuentran en el interior de la curva sobre la inmersión, y por otro lado cambiar los que quedan fuera. En la Figura 2.27 la curva  $C_1$  da lugar a una asignación ortogonal realizable si se cambian las direcciones de los segmentos interiores, como se muestra en la inmersión ortogonal de 2.28 (a). Por el contrario el cambio de los exteriores determina una inmersión no realizable (Figura 2.28 (b)). En la misma inmersión ortogonal, la curva  $C_2$  (Figura 2.29) es realizable si se modifica la dirección de los segmentos exteriores, obteniendo la asignación ortogonal realizable del gráfico 2.28 (a). Notemos que el interior de  $C_1$  coincide con el exterior de  $C_2$ , y viceversa.

De los resultados anteriores podemos obtener un proceso para determinar, dada una inmersión cilíndrica de un 4-grafo, una asignación ortogonal realizable óptima, es decir una inmersión ortogonal con el mínimo número de codos. Sin embargo este proceso no conduce a un algoritmo efectivo, puesto que para aplicar únicamente ciclos aumentadores de coste negativo realizables tenemos que considerar todos los existentes en la red bajo el flujo ortogonal en cuestión, y comprobar si son realizables o no. El número de ciclos aumentadores de coste negativo es exponencial en el número de vértices de la red, luego el proceso descrito sería exponencial. Podemos reducir el número de codos pero no obtenemos en tiempo polinomial la inmersión ortogonal óptima. Este inconveniente nos conduce a buscar otras vías que nos proporcionen inmersiones

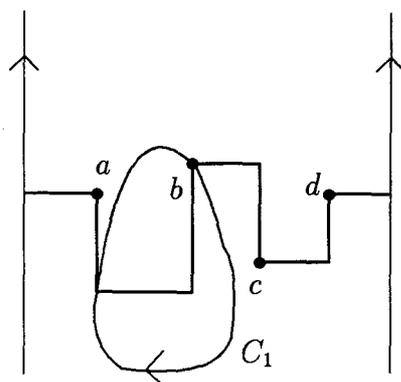


Figura 2.27:  $C_1$  es la curva asociada a un ciclo aumentador de coste negativo realizable y no esencial.

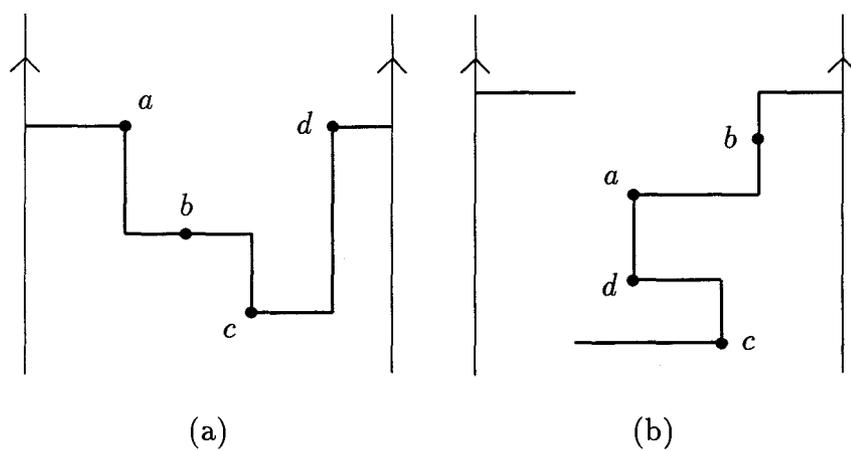


Figura 2.28: Asignaciones ortogonales obtenidas de la aplicación de  $C_1$ : (a) realizable al cambiar la dirección de los segmentos interiores, (b) no realizable al cambiar los exteriores.

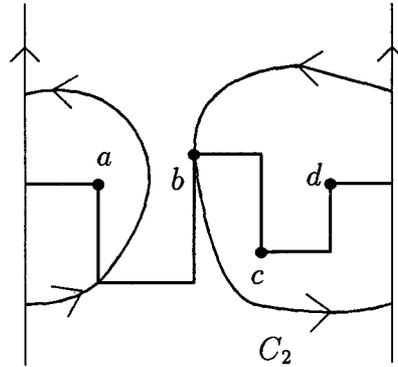


Figura 2.29: La curva  $C_2$  también es realizable si cambiamos las direcciones de los segmentos exteriores.

ortogonales con pocos codos, que en la práctica son válidas aunque no se alcance el mínimo. Además, la existencia de muchos codos en una misma arista hace no atractiva la inmersión ortogonal, como puede ocurrir en la óptima, por lo que se recurre a métodos que proporcionen inmersiones con pocos codos en las aristas. Bajo este enfoque abordamos la siguiente sección.

## 2.3 Aproximación en el número de codos

Nos planteamos en esta sección el problema de obtener inmersiones ortogonales cilíndricas que, aunque no proporcionen el mínimo número de codos, presenten una buena aproximación de él, siguiendo dos líneas distintas, una global en el grafo y otra local en cada arista.

Dada una inmersión cilíndrica de un grafo 4-plano diseñamos un algoritmo polinomial que proporciona una inmersión ortogonal equivalente a la dada, la cual aproxima el número mínimo de codos entre todas las inmersiones ortogonales equivalentes, de forma global en el grafo.

Bajo el segundo enfoque local, en primer lugar presentamos resultados teóricos que garantizan la existencia de inmersiones ortogonales cilíndricas de grafos, con  $l$  codos en cada arista,  $0 \leq l \leq 2$ , en función de la valencia del grafo. Estos hechos constituyen una mejora sobre lo obtenido en el plano por Liu, Morgana y Simeone [52]. En segundo lugar, en cuanto a aspectos computacionales, el mismo algoritmo lineal dado para el plano [53] es válido en el cilindro, proporcionando inmersiones ortogonales equivalentes a partir de representaciones cilíndricas con menos restricciones que en el plano.

### 2.3.1 Globalmente

El algoritmo que describimos a continuación nos sirve de herramienta para obtener inmersiones ortogonales cilíndricas equivalentes con pocos codos. Pueden aparecer casos en los que se obtenga justamente el mínimo número de codos, aunque en general las inmersiones obtenidas no van a ser óptimas.

El algoritmo ASIG-OPTI proporciona una asignación ortogonal óptima asociada a una inmersión cilíndrica dada. Si ésta es realizable para alguna dirección de sus segmentos, lo que se obtiene es una inmersión óptima en la malla. En caso contrario presentamos un método de obtención de “buenas” inmersiones ortogonales.

Vamos a definir un tipo especial de grafo dual a partir de una asignación ortogonal con segmentos dotados de dirección. Sea  $P$  una inmersión cilíndrica de un 4-grafo  $G$  y  $H$  una asignación ortogonal asociada. Si asociamos a un segmento arbitrario de  $H$  la dirección  $d$ , donde  $d$  puede ser vertical u horizontal, ésta determina junto con la asignación ortogonal la dirección del resto de los segmentos. Definimos el *digrafo dual*  $H_d^*$  relativo a los segmentos dirigidos de  $H$  de la siguiente forma: los vértices de  $H_d^*$  representan las caras de  $P$  y por cada segmento compartido por dos caras de la inmersión, el grafo dual contiene una arista entre los vértices correspondientes a dichas caras. Cada arista del dual lleva asociada la dirección de su segmento correspondiente en

la asignación, por lo que podremos hablar en  $H_d^*$  de aristas horizontales y verticales. Notemos que  $H_d^*$  es el dual geométrico de  $P$  relativo a segmentos, en lugar de aristas, con direcciones asignadas.

Los vértices del dual correspondientes a las caras externas ocupan un lugar distinguido, por lo que los denotamos  $A$  y  $B$ , cara inferior y superior, respectivamente. El siguiente lema trata sobre la conexión existente entre ellos.

**Lema 2.3.1.** *Existe un camino en  $H_d^*$  formado por aristas horizontales que conecta los vértices  $A$  y  $B$ .*

**Demostración:** Partimos de la asignación ortogonal cilíndrica dirigida  $H$  asociada a  $P$  y desarrollamos el cilindro por una generatriz que no contenga ningún segmento vertical. Esta operación origina la división de ciertos segmentos de  $H$  por vértices o puntos. En el primer caso duplicamos el vértice para obtener uno en cada generatriz de la representación plana, y en el segundo caso hacemos lo mismo añadiendo dos vértices nuevos al grafo. Notemos que estos vértices afectados por la operación tienen valencia uno, luego hemos obtenido en el plano una asignación o representación ortogonal  $H'$  de un grafo 4-plano. Las asignaciones de dirección de los segmentos de  $H$  no se modifican al pasar a  $H'$ .

Aplicamos el algoritmo FIND LENGTHS [64] a  $H'$  siguiendo las direcciones de sus segmentos y obtenemos una inmersión ortogonal plana en tiempo  $O(n' + b)$ , donde  $n'$  es el número de vértices de  $H'$  y  $b$  el número de codos.

Las caras externas de  $P$  están contenidas en la cara externa de la inmersión plana asociada a  $H'$ , la inferior situada bajo la inmersión ortogonal obtenida y la superior sobre ella. Trazamos una recta vertical  $r$  en el plano, que no contenga a ningún segmento vertical y que divida a la inmersión en dos partes, una a la izquierda y otra a la derecha de  $r$ . Se presenta un caso degenerado en el que esta operación no es posible, correspondiente a una inmersión de

un ciclo esencial y a una asignación que impone la dirección vertical de todos los segmentos del ciclo. Sin embargo, este tipo de asignación corresponde al caso no factible que presentaba la Figura 2.2 (a).

La existencia de  $r$  determina un camino en  $H_d^*$  entre  $A$  y  $B$ , formado por aristas horizontales en el dual. Estas aristas son las correspondientes en  $H_d^*$  a los segmentos de  $H$  que intersecta  $r$ . Obtenemos así el resultado deseado.

■

Pasamos a la descripción del algoritmo GLOBAL que nos proporciona una inmersión ortogonal con pocos codos, estudiando a continuación su corrección y complejidad. La entrada estará compuesta por una inmersión cilíndrica  $P$  de un 4-grafo y la salida será una inmersión ortogonal en el cilindro, equivalente a  $P$ , con, a lo más,  $b + 2a$  codos, donde  $b$  es el mínimo número de codos de una asignación ortogonal óptima  $H$  asociada a  $P$ , y  $a$  es la menor longitud de los caminos en  $H_d^*$  que conectan  $A$  y  $B$  y están formados por segmentos de igual dirección. La idea básica del algoritmo es, en caso de que la asignación ortogonal óptima no sea realizable, desarrollar el cilindro por una generatriz, minimizando el número de segmentos horizontales intersectados, para obtener una inmersión ortogonal plana y después añadir, a lo más, dos codos en cada segmento intersectado, para salvar la posible diferencia de altura.

### Algoritmo GLOBAL

**Entrada:** Inmersión cilíndrica  $P$  de un 4-grafo  $G$ .

**Salida:** Inmersión ortogonal cilíndrica equivalente a  $P$  con  $b + 2a$  codos.

**Paso 1.** El algoritmo recibe como entrada a la inmersión cilíndrica  $P$ .

**Paso 2.** Se aplica el algoritmo ASIG-OPTI a  $P$  obteniendo una asignación ortogonal óptima  $H$  asociada a  $P$ .

**Paso 3.** Elegimos un segmento arbitrario  $s$  de  $H$ .

**Paso 4.** Para la dirección  $d$  vertical y horizontal:

**Paso 4.1.** Dotar a  $s$  de dirección  $d$  y sentido arbitrario, convirtiendo a  $H$  en una asignación ortogonal dirigida.

**Paso 4.2.** Si  $H$  es realizable aplicar el Teorema 2.2.1 y FIN.

**Paso 4.3.** Si  $H$  no es realizable volver al Paso 4.

**Paso 5.** Para la dirección  $d$  vertical y horizontal:

**Paso 5.1.** Dotar a  $s$  de dirección  $d$  y construir el dual  $H_d^*$ .

**Paso 5.2.** Eliminar las aristas verticales del dual.

**Paso 5.3.** Determinar el camino de menor longitud entre  $A$  y  $B$  en el grafo resultante y llamar  $l_d$  a su longitud.

**Paso 6.** Sea  $a = \min\{l_v, l_h\}$ ,  $d$  la dirección para la que se ha obtenido la longitud  $a$  y  $c_d$  el camino correspondiente en  $H_d^*$ .

**Paso 7.** Se prolongan las aristas horizontales intersectadas por  $c_d$ , como indican las Figuras 2.30 y 2.31, obteniendo la generatriz  $g$ .

**Paso 8.** Se desarrolla el cilindro a lo largo de  $g$ .

**Paso 9.** Se duplican los puntos intersectados por  $g$  en  $P$ , situando en ellos vértices de valencia 1. Se obtiene una asignación ortogonal plana  $H'$ .

**Paso 10.** Se aplica el algoritmo FIND LENGTHS [64] a  $H'$  siguiendo las direcciones de sus segmentos originadas por  $d$ , obteniendo una inmersión ortogonal plana de  $H'$ .

**Paso 11.** Se vuelve a la superficie cilíndrica identificando las dos rectas verticales a que dio lugar  $g$  y eliminando los vértices añadidos.

**Paso 12.** Si sitúan dos codos en cada arista intersectada por  $g$  si los correspondientes puntos sobre las dos copias de  $g$  presentan distinta altura, según nos indica la Figura 2.32.

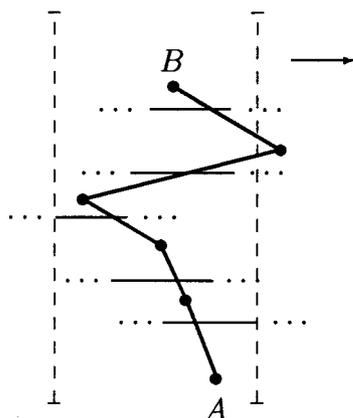


Figura 2.30: Camino  $c_d$  en  $H_d^*$  que proporciona la menor longitud entre  $A$  y  $B$ .

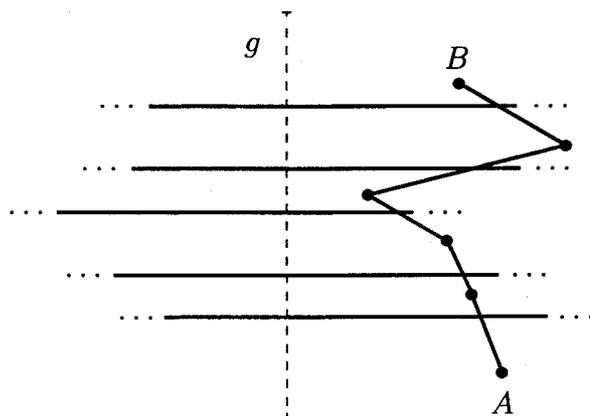


Figura 2.31: Prolongación de las aristas horizontales intersectadas por  $c_d$  y obtención de la generatriz  $g$  que corta únicamente a dichas aristas horizontales.

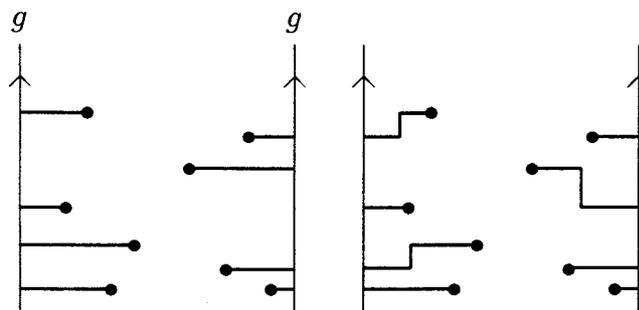


Figura 2.32: Obtención de la inmersión ortogonal cilíndrica añadiendo, en general, dos codos a cada arista cortada por  $g$ .

**Paso 13. FIN.**

Pasamos al análisis de cada uno de los pasos del algoritmo GLOBAL, demostrando la posibilidad de realización e indicando su complejidad, resultados que quedan recogidos en el siguiente teorema:

**Teorema 2.3.2.** *Dada una inmersión cilíndrica  $P$  de un 4-grafo de  $n$  vértices, el algoritmo GLOBAL proporciona una inmersión ortogonal cilíndrica equivalente a  $P$  en tiempo  $O(n^2 \log n)$ , que presenta, a lo más,  $b + 2a$  codos, donde  $b$  es el número de codos de una asignación ortogonal óptima asociada a  $P$  y  $a$  es la longitud del menor camino en el dual entre las caras externas, formado por segmentos con la misma dirección.*

**Demostración:** Para simplificar la prueba, omitiremos los pasos que se realizan en tiempo constante. La lectura de la inmersión  $P$  de entrada se realiza en tiempo lineal en  $n$ , número de vértices. Aunque sea un multigrafo, el número de aristas que presenta es de orden  $n$  por la condición de 4-grafo. El segundo paso requiere tiempo  $O(n^2 \log n)$  como indica el Teorema 1.2.9 y proporciona una asignación ortogonal óptima  $Q$  asociada a  $P$ . Denotemos por  $b$

al número de codos que contiene. El Lema 2.2.2 nos garantiza la existencia de una inmersión en la malla equivalente a  $P$  con  $4n$  codos, luego una asignación ortogonal con tal número de codos. Puesto que  $b$  es mínimo, se tiene que  $b \leq 4n$ .

El bucle que constituye el Paso 4 se realiza a lo más dos veces, una para cada dirección posible. Dentro de él, en el primer paso hay que recorrer todos los segmentos de  $Q$  para asignarles dirección y sentido a partir de  $d$ . Como  $b = O(n)$  este paso emplea tiempo lineal.

Seguidamente el algoritmo comprueba si la asignación dirigida obtenida del paso anterior es realizable. Para ello, en virtud del Teorema 2.2.1, hay que construir la inmersión asociada a  $Q$ ,  $\Gamma_h$ , y comprobar si es esférica ascendente. Lo mismo que ocurría en el paso anterior, el hecho de ser  $Q$  óptima, hace que  $\Gamma_h$  se construya en tiempo lineal. La condición de ser esférica ascendente también se comprueba en tiempo lineal [65]. En caso afirmativo, el mismo teorema citado proporciona un método lineal para obtener la inmersión ortogonal óptima correspondiente, finalizando en este caso el algoritmo.

El bucle del Paso 5 se realiza dos veces. En cada una de ellas la construcción de  $Q_d^*$  requiere tiempo lineal en el número de vértices de  $P$ , de nuevo por la condición óptima de  $Q$ . Este dual contiene tantos vértices como caras la asignación y el mismo número de aristas que ésta segmentos. En definitiva todos los valores son lineales en  $n$ . Una vez eliminadas las aristas verticales obtenemos otro grafo en el que existe un camino entre los vértices correspondientes a las caras externas, como nos indica el Lema 2.3.1 En este grafo buscamos el camino de menor longitud entre  $A$  y  $B$ , aplicando el algoritmo lineal que aparece en [54].

Una vez obtenido el menor camino para ambas direcciones, y determinado el valor  $a$ , el proceso indicado en el Paso 7 (Figuras 2.30 y 2.31) se realiza también en tiempo lineal ya que el caso extremo sería aquél en el que todas las aristas de  $Q$  fueran horizontales y  $c_d$  intersectara a todas.

El algoritmo FIND-LENGTHS [64] aplicado a  $Q'$  emplea tiempo  $O(n'+b)$ , donde  $b$  es el número de codos de  $Q$  que no varía al pasar a  $Q'$ . El valor  $n'$  es el número de vértices de  $Q'$ , que en general, el Paso 9 hace que sea mayor que el de  $Q$ . El número de vértices añadidos depende de las aristas que intersecta  $g$ , y por cada una de ellas a lo más dos vértices nuevos. En definitiva  $n' = O(n)$  y como con  $b$  ocurre lo mismo, el Paso 10 del algoritmo GLOBAL requiere tiempo  $O(n)$ .

El tiempo necesario para realizar los pasos restantes viene dado, de nuevo, en función de las aristas intersectadas por  $g$ .

Concluimos, por tanto, que el algoritmo GLOBAL proporciona una inmersión ortogonal cilíndrica equivalente a  $P$  en tiempo  $O(n^2 \log n)$ , que presenta  $b + 2a$  codos, como máximo, donde  $b$  es el número de codos de una asignación ortogonal óptima asociada a  $P$  y  $a$  es la longitud del menor camino en el dual entre las caras externas, formado por segmentos con la misma dirección. ■

Notemos que si al algoritmo GLOBAL finaliza en el Paso 4.2, la inmersión ortogonal obtenida es óptima; correspondería al caso en que la asignación ortogonal óptima fuera realizable. En caso contrario se aumenta el número de codos en  $2a$ , donde  $a$  es mínimo en el proceso realizado.

Como hemos comentado anteriormente, en general este algoritmo no proporciona el mínimo número de codos entre todas las asignaciones ortogonales realizables. Un ejemplo de ello lo constituye la asignación mostrada en la Figura 2.33 (a). Dicha asignación es óptima puesto que no presenta codos. La generatriz por la que hay que desarrollar intersecta a dos aristas horizontales, siendo este el valor mínimo de  $a$  puesto que la inmersión contiene dos ciclos esenciales disjuntos. La Figura 2.33 (b) muestra la diferencia de altura que hay que salvar añadiendo dos codos. La inmersión ortogonal resultante es la de la Figura 2.34 (a), pero en el gráfico (b) de la misma, se muestra una inmersión ortogonal sin codos de la misma inmersión cilíndrica.

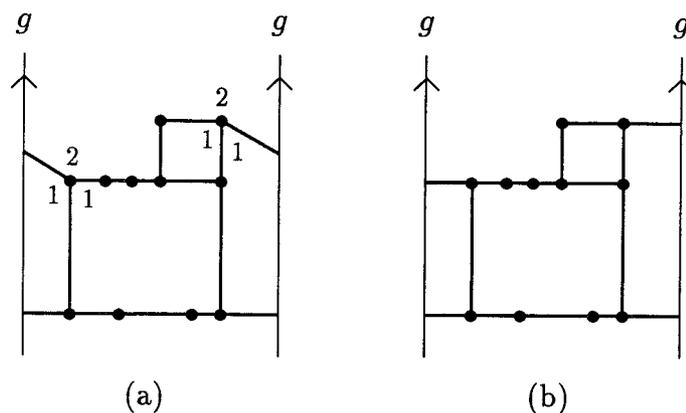


Figura 2.33: Asignación ortogonal óptima en la que es dividida una arista horizontal.

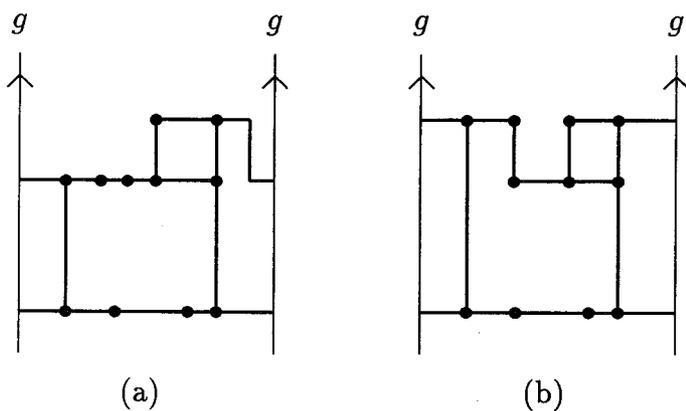


Figura 2.34: (a) Inmersión ortogonal obtenida por el algoritmo GLOBAL con la asignación y generatriz de la Figura 2.33, presentando dos codos. (b) Inmersión ortogonal óptima sin codos.

### 2.3.2 Localmente

El algoritmo GLOBAL proporciona inmersiones ortogonales cilíndricas minimizando el número de codos de asignaciones ortogonales y de aristas cortadas al desarrollar el cilindro. Estos valores de optimización son globales en la inmersión; en contraposición, en esta sección abordamos el tema centrándonos en el número de codos que contiene cada arista. Este es un enfoque local en el grafo, que computando el número de total de aristas nos conduce también a un resultado global.

Por otra parte, en el estudio realizado hasta ahora se ha partido de una inmersión cilíndrica, y no únicamente de un grafo. Los problemas sobre inmersiones ortogonales cambian sustancialmente si se parte de un grafo o de una inmersión. Tamassia [64] diseña un algoritmo polinomial que proporciona una inmersión ortogonal plana con el mínimo número de codos, equivalente a una dada como entrada. Sin embargo, si solamente partimos de un grafo 4-plano como dato, este autor junto con Garg [31] demuestran la naturaleza NP-dura del problema análogo.

Todas estas posibilidades nos dan juego para resolver problemas intermedios entre ambos planteamientos. Puesto que no se puede pretender partir de un 4-grafo y obtener la inmersión plana con el mínimo número de codos, se plantea el problema de determinar alguna inmersión del grafo que presente pocos codos, en función de los existentes en cada arista. En esta línea Liu, Morgana y Simeone presentan resultados [52] en un contexto plano. Nuestro objetivo consiste en generalizar la situación a la superficie cilíndrica, alcanzando mejoras con respecto al plano, en el sentido de obtención de un menor número de codos. Demostramos que todo grafo 4-plano admite una inmersión ortogonal cilíndrica con, a lo más, dos codos en cada arista, pudiéndose reducir este número si se disminuye la valencia de los vértices del grafo.

En la misma línea, estos últimos autores presentan un algoritmo lineal [53] que proporciona inmersiones ortogonales planas de grafos 2-conexos.

Este mismo proceso es aplicable en el cilindro, por lo que obtenemos un algoritmo lineal que construye inmersiones ortogonales cilíndricas equivalentes, con dos codos en cada arista, a partir de inmersiones cilíndricas 2-conexas no planas. Notemos que en nuestro caso podemos partir de una inmersión y obtener otra equivalente en el cilindro, mientras que en el caso plano se requiere solamente un grafo como entrada.

Necesitamos algunos conceptos previos antes de plantear y abordar el problema. Decimos que un 4-grafo cilíndrico es *k-ortogonal* si admite una inmersión ortogonal cilíndrica en la que cada arista presenta a lo más *k* codos. A tal inmersión se la llamará *k-ortogonal* en el cilindro.

Vamos a llamar  $\Pi_8$  al grafo mostrado en la Figura 2.35, también conocido como *octaedro*. Notemos que es un grafo 4-valente de 6 vértices formado por ciclos de longitud tres. A este tipo de ciclos los llamaremos *triángulos*.

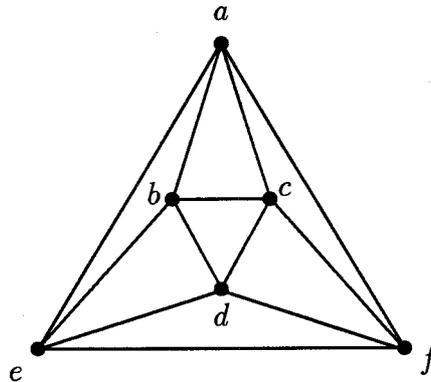


Figura 2.35: Octaedro.

$\Pi_4$  denotará al grafo completo  $K_4$  o tetraedro (Figura 2.36), grafo cúbico o 3-valente de 4 vértices también formado por ciclos de longitud tres.

Liu, Morgana y Simeone [52] demuestran el siguiente resultado para inmersiones planas de grafos:

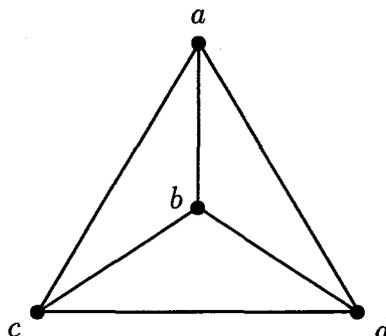


Figura 2.36 : Tetraedro.

**Teorema 2.3.3.** [52] *Un grafo  $k$ -plano,  $1 \leq k \leq 4$ , es  $(k - 1)$ -ortogonal en el plano.*

En un contexto cilíndrico, el resultado equivalente supone una mejora con respecto al plano, el cual queda recogido en el siguiente teorema:

**Teorema 2.3.4.** *Un grafo plano con vértices de valencia a lo más  $k$ ,  $2 \leq k \leq 4$ , es  $(k - 2)$ -ortogonal en el cilindro. Si el grafo es 1-valente entonces es 0-ortogonal en el cilindro.*

**Demostración:** Denotemos por  $G$  al grafo 4-plano de entrada. Si es 1-valente y lo suponemos conexo entonces está formado por una única arista. Si no fuera conexo, por varias aristas aisladas. En cualquier caso es trivial ver que admite una inmersión en el cilindro sin codos.

Para  $k = 2$ ,  $G$  es un camino o un ciclo, en el caso de que sea conexo. Si es un camino admite inmersión sin codos en el plano y en el cilindro. Por el contrario, si es un ciclo únicamente admite inmersión sin codos en el cilindro, como ciclo esencial.

En el caso en el que  $G$  sea un 3-grafo, sabemos por [52], que admite una inmersión en el plano con, a lo más, un codo en cada arista si y sólo si no es  $\Pi_4$ . Luego para los grafos no isomorfos al tetraedro tenemos garantizada la existencia de una inmersión 1-ortogonal en el cilindro. La Figura 2.37 demuestra que para  $\Pi_4$  también es posible.

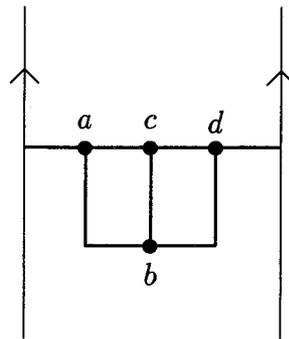


Figura 2.37: Inmersión 1-ortogonal del tetraedro en el cilindro.

Por lo tanto, todo 3-grafo admite una inmersión 1-ortogonal en el cilindro.

Para concluir la prueba tenemos que considerar el caso en el que  $G$  sea un 4-grafo plano. De nuevo se demuestra en [52] que  $G$  es 2-ortogonal en el plano si y sólo si no es isomorfo a  $\Pi_8$ . Al pasar a inmersiones cilíndricas, todos los grafos que admitían una en el plano también la admiten en el cilindro, y para la excepción constituida por el octaedro, la Figura 2.38 muestra una inmersión 2-ortogonal de  $\Pi_8$ .

■

Este teorema nos proporciona inmersiones ortogonales cilíndricas con a lo más dos codos en cada arista. Necesitamos ver que este resultado es óptimo,

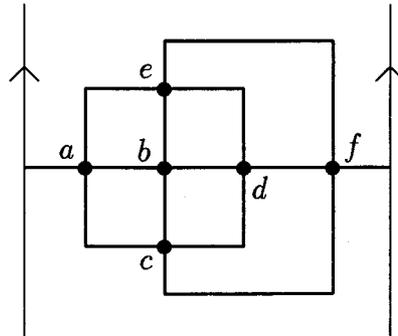


Figura 2.38: El octaedro es 2-ortogonal en el cilindro.

es decir encontrar un grafo que no admita inmersiones 1-ortogonales. Para ello presentamos el siguiente lema.

**Lema 2.3.5.** *Toda inmersión ortogonal cilíndrica de un 4-grafo con alguna cara exterior formada por vértices de valencia cuatro, presenta alguna arista con, al menos, dos codos.*

**Demostración:** Supongamos una inmersión ortogonal en las condiciones del lema. Del Polo Sur al Polo Norte, realizamos un barrido en el cilindro trasladando un círculo máximo, hasta encontrar un primer vértice de la inmersión,  $v$ , que sería el vértice de menor altura. Podemos suponer que es la cara inferior la formada por vértices de valencia 4; de lo contrario se realiza el barrido de Norte a Sur. Este vértice pertenece a la cara externa inferior, luego tiene valencia 4, por lo que todas sus direcciones están saturadas. En particular, la dirección inferior a  $v$  estará ocupada por una arista con al menos dos codos, por ser  $v$  el primer vértice encontrado en el barrido.

Esto implica que la inmersión no puede ser 1-ortogonal. La Figura 2.39 clarifica la situación. ■

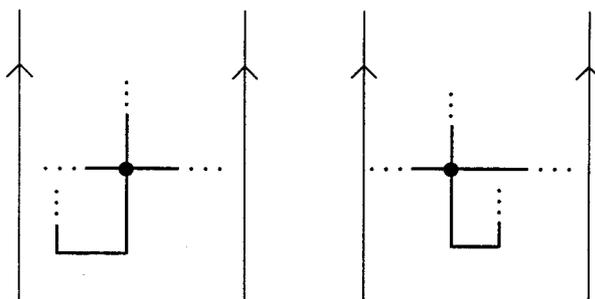


Figura 2.39: La arista  $e$  contiene al menos dos codos.

Ya sabemos que el octaedro en el plano no admite inmersiones 2-ortogonales, luego tampoco 1-ortogonales. Sus posibles inmersiones cilíndricas, propiamente dichas, se obtienen eligiendo dos caras como caras externas, bien disjuntas, compartiendo un vértice o una arista (ver Figura 2.35). En cualquiera de los casos, la frontera de las caras externas estarían constituidas por triángulos formados por vértices de valencia 4. El Lema 2.3.5 garantiza que las inmersiones ortogonales cilíndricas de  $\Pi_8$  presentan aristas con al menos dos codos. Se concluye así que  $\Pi_8$  no es 1-ortogonal, y por tanto que el Teorema 2.3.4 es óptimo. Notemos que en el caso del octaedro, como ambas caras externas estarían constituidas por vértices de valencia 4, cualquier inmersión ortogonal cilíndrica de él contiene dos o más aristas con dos o más codos.

Puesto que si un grafo contiene un subgrafo que no es  $k$ -ortogonal, entonces el grafo tampoco lo es, nos planteamos una línea de estudio para determinar que grafos son mínimos en este sentido. Decimos que un grafo no  $k$ -ortogonal es *mínimo* si no contiene subgrafos propios que tampoco lo sean.

En este contexto nos centramos en el octaedro, del que sabemos que no es 1-ortogonal. Para determinar si es mínimo tenemos que comprobar la naturaleza de sus subgrafos. El menor subgrafo que contiene sería el obtenido al eliminar una arista, y como todas las aristas juegan el mismo papel en el

grafo podemos eliminar una cualquiera. El subgrafo obtenido eliminando la arista  $(a, f)$  es 1-ortogonal tal y como muestra la Figura 2.40, luego el octaedro es mínimo para la 1-ortogonalidad.

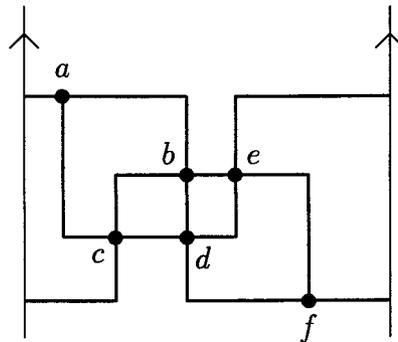


Figura 2.40: Inmersión 1-ortogonal de un subgrafo propio del octaedro.

Como hemos comentado anteriormente, en toda inmersión ortogonal de  $\Pi_8$  en el cilindro aparecerán dos aristas con dos o más codos. Por tanto el grafo resultante de subdividir una arista del octaedro sigue siendo mínimo para la 1-ortogonalidad.

Queda abierta la línea de investigación sobre inmersiones  $k$ -ortogonales en el cilindro, teniendo caracterizado el valor mínimo de  $k$ ,  $k = 2$ , para el que todo 4-grafo es  $k$ -ortogonal. El estudio del octaedro y del grafo que resulta al subdividir una arista, como grafos mínimos, da lugar a un problema abierto, como es el caracterizar los grafo mínimos para la 1-ortogonalidad.

Pasamos seguidamente al aspecto computacional del problema. A partir de los resultados teóricos anteriores y para grafos 2-conexos, el algoritmo LOCAL origina inmersiones 2-ortogonales cilíndricas equivalentes en tiempo lineal cuando recibe como entrada inmersiones en el cilindro de grafos 2-conexos. Constituye una traducción del que aparece en [53] para el plano, sin embargo en nuestro caso se puede partir de una inmersión cilíndrica y conservarla al

pasar a la malla. El número de codos que se obtiene es dos en cada arista, mejorando el resultado plano de tres codos en cada una.

### Algoritmo LOCAL

**Entrada:** Inmersión cilíndrica no plana  $P$  de un 4-grafo  $G$  2-conexo.

**Salida:** Inmersión 2-ortogonal cilíndrica equivalente a  $P$ .

**Paso 1.** El algoritmo recibe como entrada la inmersión cilíndrica  $P$  de  $n$  vértices.

**Paso 2.** Se define una st-numeración de  $G$  de forma que:

- Los vértices 1 y 2 pertenezcan a la cara externa inferior.
- Los vértices  $n - 1$  y  $n$  pertenezcan a la cara externa superior.
- Los vértices 1 y  $n$  no sean adyacentes.
- Los pares  $(1, 2)$  y  $(n - 1, n)$  no constituyan pareja de corte de  $G$ .

**Paso 3.** La inmersión 2-ortogonal se construye considerando en cada vértice las estructuras que muestra la Figura 2.41, en función de las aristas entrantes o salientes que define la st-numeración. Si el vértice en curso no es 4-valente se toman solamente las aristas incidentes que presente.

La corrección y complejidad del algoritmo LOCAL vienen dados por el siguiente teorema:

**Teorema 2.3.6.** *Dada una inmersión cilíndrica  $P$  de un 4-grafo 2-conexo de  $n$  vértices, el algoritmo LOCAL proporciona una inmersión 2-ortogonal cilíndrica equivalente a  $P$  en tiempo  $O(n)$ .*

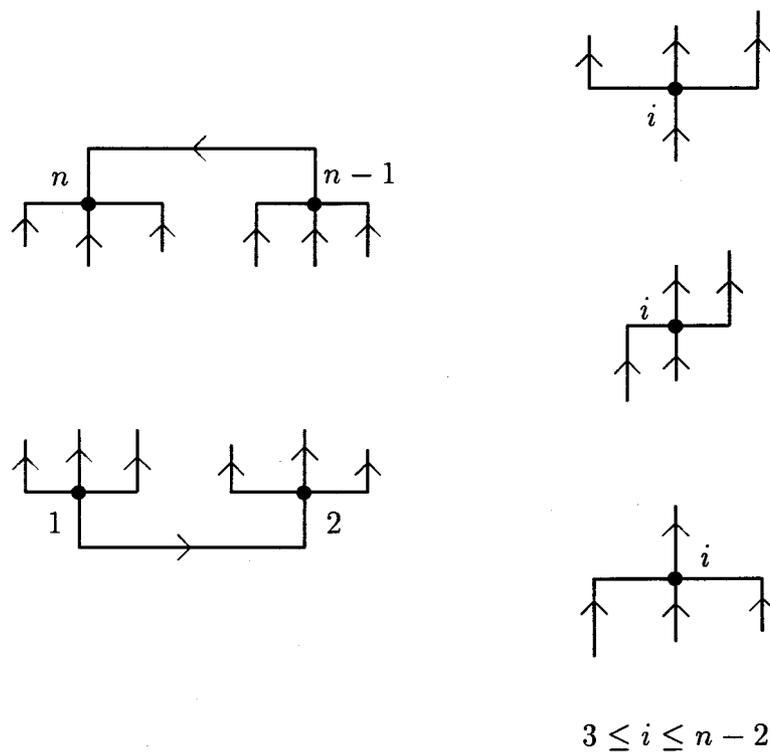


Figura 2.41: Estructuras de las aristas incidentes en cada vértice de la inmersión.

**Demostración:** La lectura de  $P$  es lineal en  $n$  puesto que se trata de una inmersión cilíndrica. La 2-conexión del grafo y los resultados de [53] hacen posible la existencia de una  $st$ -numeración con las condiciones dadas. El hecho de que la arista  $(1, n)$  no pertenezca a la inmersión no constituye un problema, ya que podemos añadirla como arista virtual, aunque la inmersión deje de ser cilíndrica. La obtención de tal  $st$ -numeración se realiza en tiempo lineal.

Por último, las estructuras consideradas son las mismas que en el caso plano, luego la corrección de este paso se remite a [53]. ■

## 2.4 Conclusiones y problemas abiertos

La línea de investigación sobre inmersiones ortogonales ha adquirido una gran importancia en los últimos años dentro del área de representación de grafos. La tónica general en los trabajos existentes ha sido un planteamiento plano de los problemas, apareciendo alguno sobre ortogonalidad en  $\mathbf{R}^3$ . Nuestro propósito ha sido la generalización a otras superficies, dedicando esta parte de la memoria al caso del cilindro.

En el capítulo anterior nos centramos en las asignaciones ortogonales, y en este hemos analizado la forma de pasar a la malla cilíndrica, dependiendo de la dirección sobre los segmentos. La conclusión más evidente es la gran diferencia que surge en el problema, con relación al caso plano. Tamassia [64] prueba que toda asignación o representación ortogonal plana equivale a una inmersión, independientemente de la dirección considerada. Puesto que en el cilindro este hecho no ocurre, hemos introducido y caracterizado las asignaciones ortogonales realizables.

El propósito del trabajo de Tamassia, así como el nuestro, consiste en determinar una inmersión ortogonal equivalente a la dada, con el mínimo número de codos. Para el plano, es posible en tiempo  $O(n^2 \log n)$  y para el cilindro hemos caracterizado teóricamente tal inmersión. Sin embargo, este

resultado no nos conducido a un algoritmo eficiente. Por tanto, queda como principal problema abierto, el diseño de un algoritmo, probablemente utilizando otras técnicas, que sea capaz de proporcionar en tiempo polinomial una inmersión ortogonal cilíndrica óptima.

Como alternativa, presentamos otras vías de resolución que, aunque no obtengan el resultado óptimo, se aproximen a la solución. En un primer resultado, bajo nociones de representaciones de visibilidad, se obtienen en tiempo lineal, inmersiones ortogonales equivalentes a las dadas, con  $4n$  codos, siendo  $n$  el número de vértices del grafo. El algoritmo GLOBAL mejora este resultado aunque emplea tiempo  $O(n^2 \log n)$ .

El planteamiento considerado en los problemas anteriores ha sido global en el grafo en cuanto al número de codos. Esto puede originar pocos codos en la inmersión total, pero también aristas con un número elevado de codos, situación que no es deseable. Por este motivo se ha abordado una línea de estudio local en cada arista. Hemos denotado grafos  $k$ -ortogonales a aquellos que admiten una inmersión ortogonal cilíndrica con, a lo más,  $k$  codos en cada arista. Todo grafo 4-plano es 2-ortogonal en el cilíndrico. El octaedro es un ejemplo de grafo que requiere la existencia de aristas con dos codos. Como problema abierto permanece el de determinar todos los grafos que se encuentran en esta situación, es decir caracterizar los grafos mínimos para la 1-ortogonalidad. Se presenta en último lugar el algoritmo LOCAL para grafos 2-conexos que proporciona en tiempo lineal inmersiones 2-ortogonales equivalentes a las dadas. Este algoritmo hace uso de una st-numeración del grafo, luego es necesaria la 2-conexión. Queda pendiente el desarrollo de otros métodos para obtener en tiempo eficiente inmersiones 2-ortogonales cilíndricas de grafos cualesquiera.

No hemos considerado en nuestro trabajo el estudio del área que ocupan las inmersiones ortogonales cilíndricas obtenidas. Para el cilindro la noción de área tiene que ser diferente a la del plano puesto que la estructura topológica de aquél, hace que los círculos máximos recorran, digámoslo de alguna forma,

toda la “anchura” del cilindro. La dimensión a determinar sería la altura de los trazados ortogonales.

## **Parte II**

# **Inmersiones ortogonales en superficies**

Tal y como se ha comentado repetidas veces a lo largo de esta memoria, una de las principales motivaciones para el estudio de las inmersiones ortogonales de grafos es el diseño de circuitos impresos. Naturalmente, por cuestión de simplicidad, los estudios teóricos han comenzado con el plano, sin embargo, en la práctica los circuitos impresos no suelen corresponderse con grafos planos, sino con grafos en superficies de mayor género. Es por ello que se hace necesario el estudio teórico de inmersiones ortogonales de grafos en superficies. En la primera parte de esta memoria se inició dicha tarea, considerando el caso no plano más simple como es el cilindro; en esta segunda parte nos centramos en inmersiones de grafos en superficies de cualquier género.

Para poder hablar de inmersión ortogonal en una superficie  $S$  cualquiera necesitamos en primer lugar, una representación de  $S$  adecuada en la que se pueda generalizar los conceptos de vertical y horizontal, definiendo así la noción de malla en  $S$ . Las representaciones ortogonales estándares de las superficies presentan esta característica y es con ellas con las que trabajamos al tratar inmersiones ortogonales.

Con el objetivo de minimizar el número de codos, el primer intento se centraría en obtener inmersiones ortogonales óptimas a partir de grafos o representaciones dados. Bajo esta línea general aparecen situaciones en las que el problema es de naturaleza NP-dura.

Una inmersión de un grafo en una superficie cualquiera, si aquella es plana, no contiene ningún ciclo esencial. La generalización que nos proponemos nos lleva a considerar inmersiones no planas, en las que hay que distinguir una parte esencial y otra no esencial o plana. En el Capítulo 3 introducimos el concepto de inmersiones esencialmente equivalentes en superficies, el cual nos informa sobre la posible equivalencia entre ambas partes por separado, y demostramos que el problema de *la inmersión ortogonal óptima esencialmente equivalente* es de naturaleza NP-dura: dada una inmersión  $\phi$  de un 4-grafo en una superficie  $S$ , el problema de encontrar una inmersión ortogonal en  $S$ , esencialmente equivalente a  $\phi$  es de naturaleza NP-dura.

Si en lugar de partir de una inmersión en una superficie, únicamente partimos de un grafo  $G$ , y nos preguntamos por la inmersión ortogonal óptima de  $G$  entre todas las posibles en cualquier superficie, obtenemos el problema de *la inmersión ortogonal en superficies*. En el Capítulo 4 demostramos su naturaleza NP-dura.

La intratabilidad de estos problemas motiva la búsqueda de algoritmos efectivos, que, aunque no proporcionen el mínimo número de codos, originen inmersiones “buenas”, en la línea de pocos codos. Este incentivo nos lleva en una segunda parte del Capítulo 4, a presentar dos algoritmos lineales GPCS e IPCS, que construyen inmersiones ortogonales con un cierto número de codos en cada arista. La entrada del primero está compuesta por un 4-grafo 2-conexo, en general no plano, y para el segundo se parte de una inmersión de un 4-grafo 2-conexo en una superficie.

## Capítulo 3

# Inmersiones ortogonales esencialmente equivalentes

En este capítulo presentamos el concepto de inmersiones esencialmente equivalentes para representaciones de grafos en superficies de género arbitrario. Esta noción distingue las dos componentes, esencial y no esencial, de una inmersión. Demostramos que el problema de determinar si existe una inmersión ortogonal esencialmente equivalente a una dada, con el mínimo número de codos, es de naturaleza NP-dura. Esta cuestión constituye *el problema de la inmersión ortogonal esencialmente equivalente*.

### 3.1 Introducción

Cuando el ingeniero se plantea el diseño de un circuito impreso en el que tiene que trazar unas determinadas conexiones mediante segmentos horizontales y verticales, se encuentra con el problema de cruces en la placa entre distintas líneas en puntos que no corresponden a nodos. Estos nodos, en los que se realizan las conexiones, y los trazados de éstas definen un grafo, en la mayoría

de los casos no plano. En función de cómo se dibuje este grafo en la placa, se producirá un número mayor o menor de cruces. La situación ideal sería encontrar la representación con el mínimo número de cortes entre aristas, problema conocido como “crossing number”, del que se conoce su naturaleza NP-dura [30].

Existen diversas heurísticas que proporcionan dibujos con un número aceptable de cruces. Para evitarlos, una técnica habitual en el diseño de VLSI es realizar orificios de una cara de la placa a la otra. De esta forma no se diseña el circuito en una superficie plana, sino en otra de género mayor, de forma que las ventanas de la representación ortogonal estándar de ésta evitan los cruces indeseados. Además, como es bien sabido, los trazados de las placas siguen la dirección horizontal y vertical. Estos hechos nos han conducido al estudio de inmersiones ortogonales de grafos en superficies no planas.

En el diseño puede que únicamente interese las conexiones entre los nodos, o bien que haya que tener en cuenta la posición relativa entre ellas. Estas dos posibilidades conducen a dos planteamientos; en la primera se parte del grafo para construir la placa y en el segundo caso además de éste se conoce una inmersión suya. El primer planteamiento es el considerado en el capítulo siguiente y en este trataremos con inmersiones de grafos.

Como hemos apuntado, consideramos a lo largo de estos dos capítulos las representaciones ortogonales estándar de las superficies. Una condición necesaria y suficiente para que un grafo admita una inmersión ortogonal en una superficie adecuada es que sea un 4-grafo, es decir vértices con valencias no mayores a cuatro.

Toda inmersión de un grafo en una superficie ortogonal estándar nos informa sobre las aristas que cortan a alguno de sus rectángulos, dando lugar a una clasificación de los ciclos en esenciales y no esenciales, y en el grafo entre subgrafo esencial y no esencial. Puesto que son dos partes con propiedades claramente distintas, dadas dos inmersiones de un grafo comparamos estos dos subgrafos por separado, derivando en el concepto de inmersiones esencialmente

equivalentes.

En este contexto planteamos el problema de la inmersión ortogonal óptima esencialmente equivalente en los siguientes términos: dada una inmersión de un 4-grafo en una superficie, nos preguntamos por una inmersión ortogonal esencialmente equivalente a la dada con el mínimo número de codos. Demostramos en este capítulo la naturaleza NP-dura de este problema.

## 3.2 Inmersiones rectilíneas esencialmente equivalentes

Para demostrar la naturaleza NP-dura del problema de la inmersión ortogonal óptima esencialmente equivalente, pasamos a términos de problemas de decisión y NP-completitud. En nuestro caso el problema de decisión asociado preguntaría por la existencia de una inmersión rectilínea en la superficie esencialmente equivalente a la dada. La naturaleza NP-completa de éste determina la NP-dureza del primero.

Para abordar este problema tenemos que definir, en primer lugar, el concepto de inmersión esencialmente equivalente. Recordamos que dos inmersiones de un grafo en la superficie  $O_g$  son equivalentes si sus sistemas de rotación coinciden y sus ciclos presentan la misma naturaleza en cuanto a esencial o no esencial. Esto nos lleva a distinguir en un grafo inmerso en una superficie entre los “subgrafos planos” y los ciclos esenciales que contiene; la naturaleza de estas dos partes es claramente distinta y es esta dicotomía la que nos conduce a dicho concepto. En el Capítulo 0 hablamos de inmersiones equivalentes y de conexión de grafos y a partir de ellos decimos que dos inmersiones en una superficie son *esencialmente equivalentes* si los dos subgrafos constituidos, uno por los bloques con ciclos esenciales y el otro por los bloques sin ciclos esenciales, son equivalentes dos a dos. La Figura 3.1 muestra una inmersión de un grafo en el toro, dos inmersiones de dicho grafo no esencial-

mente equivalentes a la dada se pueden contemplar en las Figuras 3.2 y 3.3 y se considera en la Figura 3.4 una inmersión del grafo esencialmente equivalente al ejemplo dado.

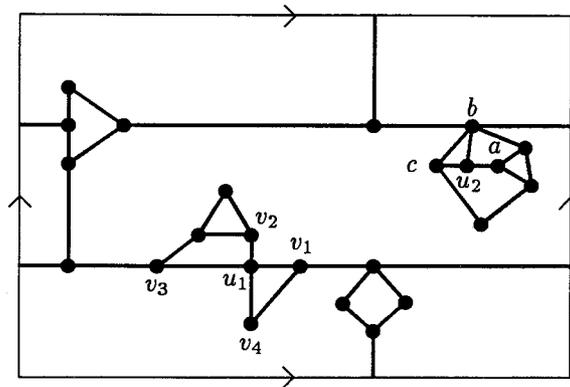


Figura 3.1: Inmersión  $\phi$  de un grafo en el toro.

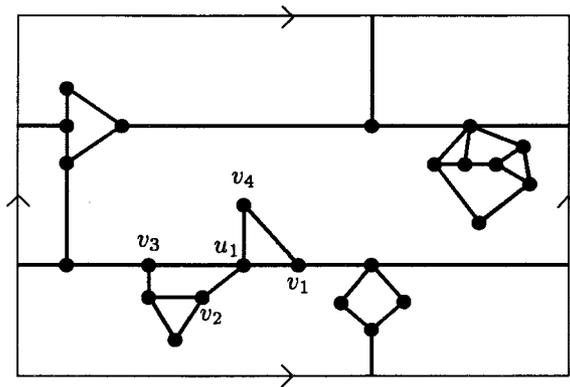


Figura 3.2: Inmersión no esencialmente equivalente a  $\phi$ .

En esta sección demostramos la naturaleza NP-dura del problema planteado y para ello consideramos en primer lugar su correspondiente problema

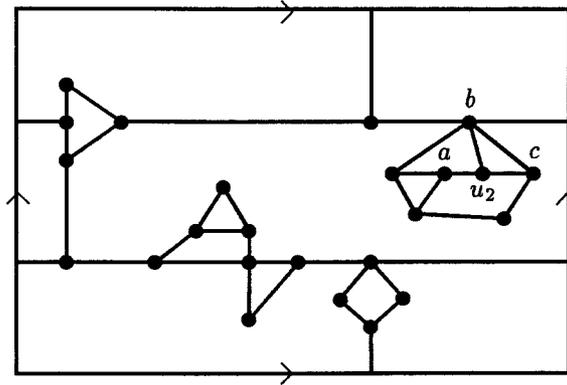


Figura 3.3: Inmersión no esencialmente equivalente a  $\phi$ .

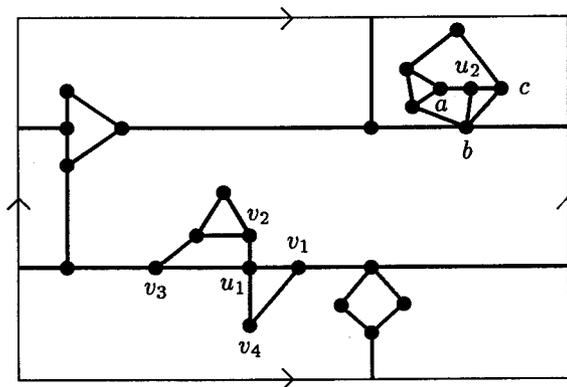


Figura 3.4: Inmersión esencialmente equivalente a  $\phi$ .

de decisión, que formulamos de la siguiente forma:

INMERSIÓN RECTILÍNEA ESENCIALMENTE EQUIVALENTE (IREE):

ENTRADA: Superficie  $S$ , 4-grafo  $G$ , inmersión  $\phi : G \rightarrow S$ .

PREGUNTA: ¿Existe una inmersión rectilínea de  $G$  en  $S$  esencialmente equivalente a  $\phi$ ?

El problema IREE es el problema de decisión asociado al de búsqueda planteado al comienzo de la sección y como hemos visto en Preliminares la naturaleza NP-completa del primero conduce de forma inmediata al carácter NP-duro del segundo. Siguiendo la forma habitual de demostración en la teoría de la NP-completitud, consideramos el problema 3-SAT, planteado en el Capítulo 0, para relacionarlo con IREE.

**Teorema 3.2.1.** *El problema de la inmersión rectilínea esencialmente equivalente (IREE) es NP-completo.*

**Demostración:** En primer lugar comprobamos fácilmente que IREE pertenece a la clase NP. Consideramos un algoritmo no determinista polinomial con las dos etapas siguientes: la etapa “guessing” origina una inmersión  $\phi'$  de  $G$  en  $S$ . Los datos  $G$ ,  $S$  y  $\phi$  del problema, junto con  $\phi'$  será la entrada de la segunda etapa o etapa “checking”, la cual verifica en tiempo polinomial en el número de aristas de  $G$  si  $\phi'$  es, por una parte, esencialmente equivalente a  $\phi$  y por otra, rectilínea. Para comprobar si  $\phi$  y  $\phi'$  son esencialmente equivalentes tenemos que localizar en primer lugar los bloques de cada grafo, operación que se puede hacer en tiempo lineal [1]. Puesto que el concepto de esencialmente equivalente distingue “parte esencial y no esencial” de una inmersión, el siguiente paso será clasificar los bloques de ambas inmersiones según si contienen ciclos esenciales o no. La primera etapa de este algoritmo no determinista ha originado una inmersión  $\phi'$  de  $G$  en  $S$ , esto es, ha proporcionado el sistema de rotación en cada vértice y ha etiquetado las aristas que cortan a algún rectángulo de la representación ortogonal estándar de  $S$  (es decir, la inmersión también nos proporciona los ciclos esenciales); de esta forma podemos considerar por un

lado los bloques “esenciales” y por otro los “no esenciales” de ambas inmersiones y comprobar si el carácter del mismo bloque, según esta clasificación, en ambas inmersiones es el mismo. Es claro que el número de bloques que contiene cada inmersión es lineal en el número de aristas o vértices del grafo, luego también lo es esta operación. Llegado a este punto el último paso del algoritmo no determinista consiste en comparar los bloques análogos de  $\phi$  y  $\phi'$  y comprobar si coinciden los sistemas de rotación en cada vértice, operación que emplea tiempo polinomial.

Para verificar si la inmersión  $\phi'$  es rectilínea basta comparar para cada arista las coordenadas de sus vértices en  $S$ , a lo más dos comparaciones por arista, y comprobar si alguna de las dos coordenadas coincide en ambos vértices. Si existe alguna arista para la cual sus dos vértices no tienen ninguna coordenada en común, entonces  $\phi'$  no es rectilínea.

La segunda parte de la demostración define una transformación polinomial del problema 3-SAT en IREE, según la notación introducida 3-SAT  $\infty$  IREE. Sea  $U = \{u_1, u_2, \dots, u_n\}$  un conjunto de variables y  $C = \{c_1, c_2, \dots, c_m\}$  un conjunto de cláusulas constituyendo una entrada arbitraria  $\mathcal{R}$  de 3-SAT. A partir de  $\mathcal{R}$  construimos una inmersión  $\phi_{\mathcal{R}}$  de un grafo  $G_{\mathcal{R}}$  en una superficie adecuada como sigue.

A cada cláusula  $c_i$ ,  $i = 1, \dots, m$ , excepto a las que contienen a un literal y a su negación, le asociamos la inmersión mostrada en la Figura 3.5, a la que denotaremos  $\phi_i$ , en la que cada subgrafo señalado en la Figura 3.6 representa uno de los tres literales de la cláusula, dispuestos en el mismo orden que éstos.

Situamos estas inmersiones en la esfera, desde  $\phi_1$  hasta  $\phi_m$  de Norte a Sur, y consideramos las variables comunes a sus correspondientes cláusulas.

Si existen dos cláusulas consecutivas que contienen al literal  $u_l$  (ó  $\bar{u}_l$ ),  $1 \leq l \leq n$ , se añade una arista entre sus subgrafos asociados como se indica en la Figura 3.7 (a). Con esta operación se construye una especie de “camino” en la inmersión entre los subgrafos asociados a los literales  $u_l$  y otro entre los

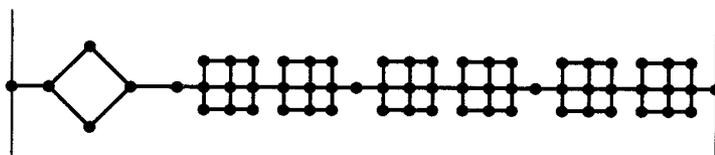


Figura 3.5: Inmersión asociada a cada cláusula.



Figura 3.6: Inmersión asociada a cada literal.

$\bar{u}_i$ . Entre los últimos literales de estos caminos se realiza la unión mostrada en la Figura 3.7 (b).

Notemos que cada subgrafo asociado a un literal se une a lo más dos veces, una con una subinmersión superior y otra con una inferior, utilizando para dichas uniones vértices distintos del subgrafo.

De esta forma hemos definido una inmersión  $\phi_{\mathcal{R}}$  de un grafo  $G_{\mathcal{R}}$ . Para evitar posibles cruces entre las aristas de  $\phi_{\mathcal{R}}$  en puntos que no sean vértices, aumentamos el género de la superficie; partiendo de la esfera consideramos  $\phi_{\mathcal{R}}$  como una inmersión de  $G_{\mathcal{R}}$  en una superficie  $S$  cuyo género, a lo más, es igual al cuadrado del número de aristas de dicha inmersión.

La transformación definida es polinomial en el número de cláusulas y variables ya que la inmersión  $\phi_{\mathcal{R}}$  está compuesta, como máximo, por  $m$  subinmersiones como la mostrada en la Figura 3.5. Cada una de las uniones entre las variables de las cláusulas (Figura 3.7) se realiza en tiempo constante, y para determinar cuáles tenemos que realizar hay que efectuar  $O(nm)$  com-

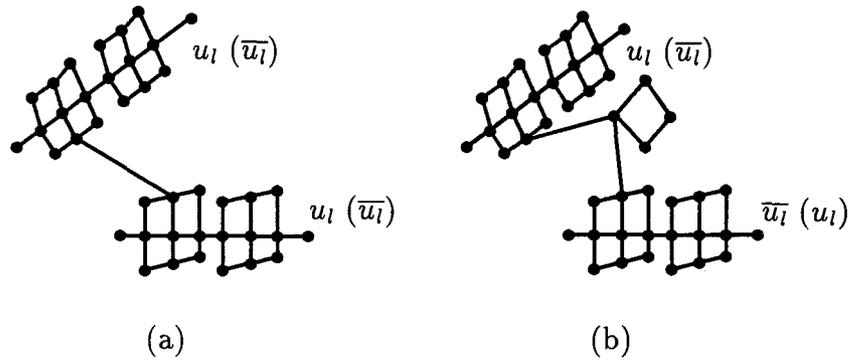


Figura 3.7: (a) Unión entre inmersiones asociadas al mismo literal.  
 (b) Unión entre inmersiones asociadas a literales opuestos.

paraciones. Por tanto hemos asociado una inmersión  $\phi_{\mathcal{R}}$  de un 4-grafo  $G_{\mathcal{R}}$  en una superficie  $S$  a cualquier entrada arbitraria del problema 3-SAT, mediante una transformación polinomial.

La Figura 3.8 ilustra un ejemplo concreto para el conjunto de variables  $U = \{u_1, u_2, u_3, u_4, u_5\}$  y de cláusulas  $C = \{c_1, c_2, c_3\}$  donde  $c_1 = \{u_2, u_3, u_5\}$ ,  $c_2 = \{u_1, \bar{u}_4, u_5\}$  y  $c_3 = \{u_2, u_3, \bar{u}_5\}$ .

Para concluir que  $3\text{-SAT} \in \text{IREE}$  nos queda probar que  $\mathcal{R}$  es satisfacible si y sólo si existe una inmersión rectilínea de  $G_{\mathcal{R}}$  esencialmente equivalente a  $\phi_{\mathcal{R}}$ .

En cualquier inmersión rectilínea de  $G_{\mathcal{R}}$  esencialmente equivalente a  $\phi_{\mathcal{R}}$ , decimos que el subgrafo asociado a un literal de una cláusula es vertical si los dos vértices de valencia 1 en él están situados sobre el mismo meridiano; y es horizontal si aquellos están situados sobre el mismo paralelo (estas son las dos únicas opciones). Con esta notación exponemos los siguientes hechos:

- Toda inmersión rectilínea de cada  $\phi_i$  contiene un subgrafo vertical ya que,

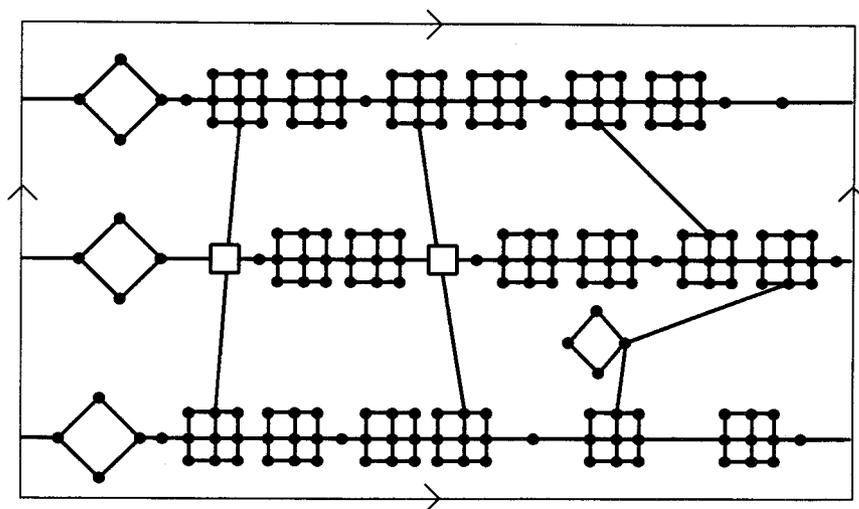


Figura 3.8: Un ejemplo para tres cláusulas.

según nos muestra la Figura 3.9, los vértices de valencia 1 que resultan al eliminar de  $\phi_i$  los subgrafos asociados a los literales, están situados en paralelos distintos.

- Los vértices de valencia 4 contenidos en los subgrafos de la Figura 3.6 y las posteriores uniones realizadas hacen que, en cualquier inmersión rectilínea de  $\phi_{\mathcal{R}}$ , dos subgrafos correspondientes a literales asociados a la misma variable sean, uno vertical y otro horizontal si y sólo si un literal es la negación del otro. Dicho de otra forma, todos los subgrafos asociados al mismo literal presentarán la misma dirección (podremos hablar de la dirección de cada literal), mientras que dos subgrafos correspondientes a literales opuestos tendrán dirección contraria.

Estos dos hechos demuestran que, dada una inmersión rectilínea en  $S$  esencialmente equivalente a  $\phi_{\mathcal{R}}$ , la siguiente aplicación sobre  $U$  es una asig-

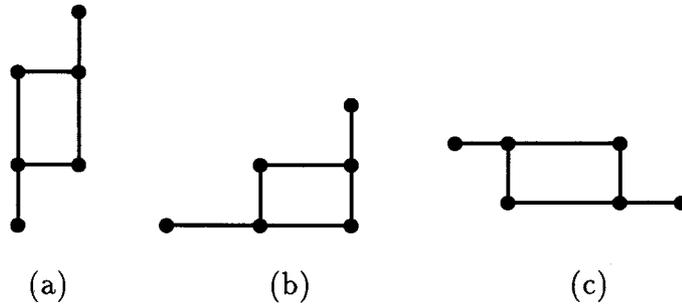


Figura 3.9: Inmersiones rectilíneas del grafo asociado a una cláusula.

nación de verdad satisfactoria para  $\mathcal{R}$ :

$$t(u_l) = \begin{cases} T & \text{si el literal } u_l \text{ es vertical} \\ F & \text{si el literal } u_l \text{ es horizontal} \end{cases}$$

donde  $T$  corresponde al valor *verdadero* y  $F$  al *falso*.

Las cláusulas que contienen a un literal y a su negación se satisfacen trivialmente y en el caso en el que alguna variable no tenga representante en ninguna cláusula, se le asigna un valor arbitrario.

Las afirmaciones anteriores han permitido la obtención de una respuesta afirmativa de 3-SAT a partir de una de igual tipo para IREE; pasamos a la segunda parte en la que, dada una asignación de verdad satisfactoria para  $\mathcal{R}$ , construimos una inmersión rectilínea de  $G_{\mathcal{R}}$  en  $S$  esencialmente equivalente a  $\phi_{\mathcal{R}}$ . El método a seguir es la inducción en el número de cláusulas de  $\mathcal{R}$ . Si  $m = 1$  es trivial y para valores pequeños de  $m$  se aplican las mismas técnicas que en el paso de inducción. En lo que sigue trataremos indistintamente al literal y a su subgrafo asociado. La idea general que guiará el proceso es la misma que la considerada en la implicación anterior, esto es, los subgrafos verdaderos presentarán dirección vertical y los falsos dirección horizontal y cada inmersión  $\phi_i$  adoptará la estructura mostrada en la Figura 3.9 (c). Se cubrirá la diferencia de altura con los subgrafos verticales y se completará



- b) Si ambos literales son verdaderos se realiza el dibujo de forma que exista un paralelo que intersecte a ambos.
- c) Análogamente para un meridiano si son falsos.
- d) Los literales verdaderos tienen que preservar el orden ascendente, es decir, dados dos literales verticales, en general de cláusulas distintas, el de índice superior tiene que contener puntos (distintos de los vértices) más altos que otros del de índice inferior. Esta condición se verifica trivialmente para los literales de la misma cláusula (ver Figura 3.10).
- e) Cada literal tiene que tener intersección no vacía y no finita con su banda correspondiente. En caso de consecutividad de literales verdaderos se realizará el dibujo en la banda asociada al de índice mayor. Este caso puede dar lugar a la única excepción del punto primero con respecto al dibujo de la siguiente cláusula.

El paso siguiente será la adecuación del subgrafo restante de  $\phi_m$  después de dibujar los literales y considerando la estructura indicada anteriormente.

El último paso del proceso será realizar las uniones entre los “caminos” asociados a los literales opuestos de cada variable, las cuales se realizan según nos indica la Figura 3.11, operación factible por las condiciones impuestas en el dibujo.

El género de  $S$  garantiza que los únicos puntos comunes entre las aristas de  $\phi_{\mathcal{R}}$  son vértices.

Claramente se observa que la inmersión obtenida es esencialmente equivalente a  $\phi_{\mathcal{R}}$ .

Con esto hemos demostrado que si  $\mathcal{R}$  es satisfacible es posible construir una inmersión rectilínea esencialmente equivalente a  $\phi_{\mathcal{R}}$ . Concluimos, por tanto, que IREE es de naturaleza NP-completa. ■

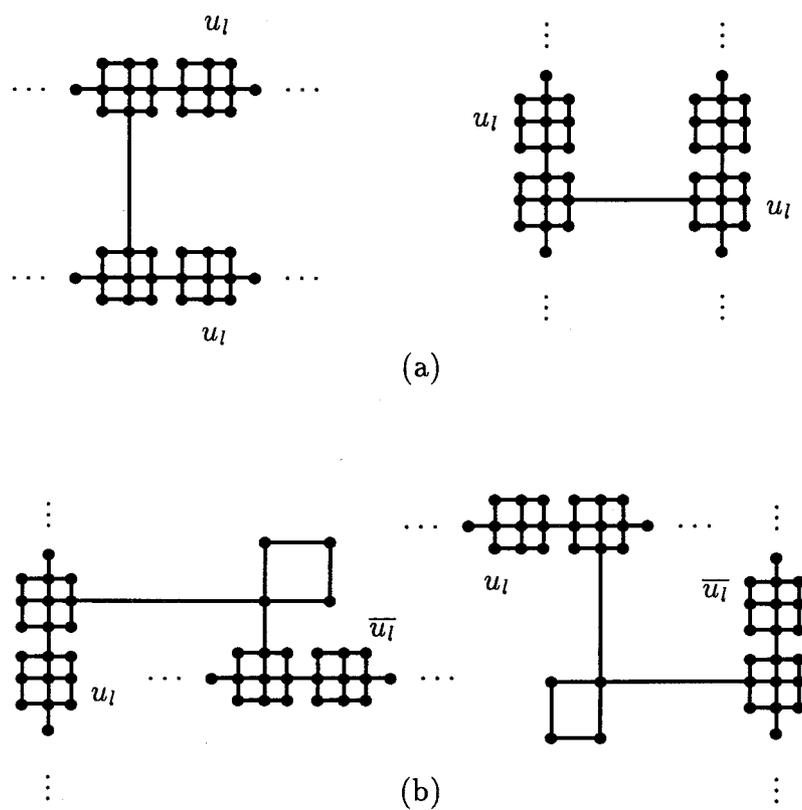


Figura 3.11 : Inmersiones rectilíneas de las uniones de la Figura 3.7.

La Figura 3.12 muestra una inmersión rectilínea esencialmente equivalente a la dada en la Figura 3.8, correspondiente a la siguiente asignación de verdad satisfactoria para el ejemplo dado:

$$t(u_1) = 0 \quad t(u_3) = 0 \quad t(u_5) = 1$$

$$t(u_2) = 1 \quad t(u_4) = 0$$

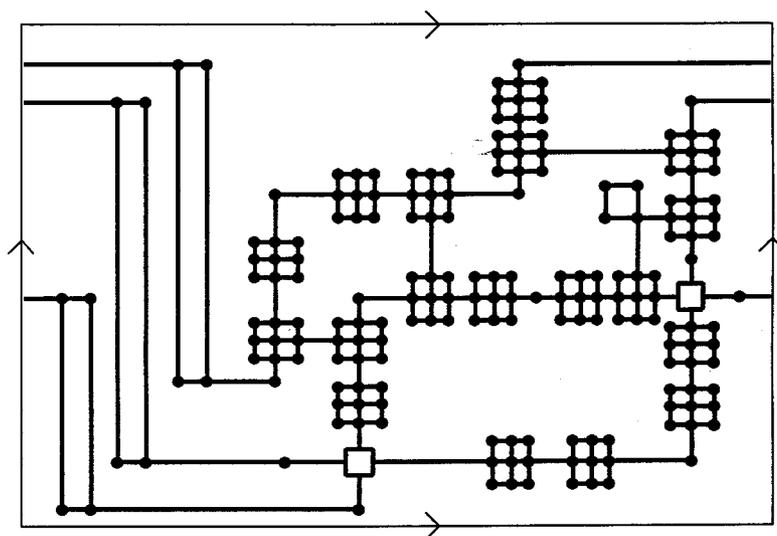


Figura 3.12: Inmersión rectilínea del grafo mostrado en la Figura 3.8.

Como apuntábamos al principio de la sección, de la NP-completitud de IREE, Teorema 3.2.1, se sigue la naturaleza NP-dura del correspondiente problema de optimización, resultado que queda recogido en el siguiente corolario:

**Corolario 3.2.2.** *Dada una inmersión  $\phi : G \rightarrow S$  de un grafo  $G$  en una superficie  $S$ , computar una inmersión ortogonal de  $G$  en  $S$  esencialmente equivalente a  $\phi$  con el mínimo número de codos es un problema NP-duro.*

### 3.3 Conclusiones y problemas abiertos

El objetivo de esta parte de la memoria ha sido iniciar el estudio sobre inmersiones ortogonales en superficies arbitrarias, generalizando el problema planteado en el plano. La motivación de esta línea se centra en el diseño de circuitos impresos, los cuales se realizan en superficies, en general, no planas.

Hemos introducido la relación de equivalencia esencial entre inmersiones en superficies, para distinguir entre subinmersión esencial y no esencial. En este contexto se tiene la naturaleza NP-dura del problema de la inmersión ortogonal óptima esencialmente equivalente.

Es claro que si dos inmersiones son equivalentes entonces también lo son en el sentido esencial, sin embargo es fácil comprobar que pueden ser esencialmente equivalentes y no equivalentes, un ejemplo de ello lo encontramos en la Figura 3.4. Este hecho hace que la naturaleza NP-completa de IREE no determine el mismo carácter para el problema análogo considerando inmersiones equivalentes. Surge de esta forma un problema abierto, para el que habría que buscar un algoritmo efectivo que proporcionara una inmersión ortogonal óptima equivalente a la dada, o bien demostrar que es NP-duro.

Este sería el problema abierto más inmediato que surge a partir de los resultados presentados, sin embargo la actualidad del tema deriva en innumerables líneas de investigación. Si en lugar de tratar con superficies en general, particularizamos en una concreta como el toro, doble toro, etc., podemos plantearnos los mismos problemas e intentar obtener algoritmos efectivos que construyan inmersiones ortogonales o rectilíneas en ellas, con la idea de minimizar el número de codos. Esta es la línea tratada en la primera parte para el cilindro.

Puesto que el carácter NP-duro de un problema no es “bueno” en el sentido computacional, podemos imponerle a la inmersión propiedades específicas, como grado de conexión, o disminuir la valencia de los vértices, para compro-

bar si bajo estas restricciones podemos obtener resultados computacionales efectivos.

Nuestro planteamiento nos impone considerar 4-grafos para que admitan inmersiones ortogonales. Podemos encontrar en el plano trabajos en los que los vértices están representados por estructuras tales como rectángulos, por ejemplo [24], lo que permite considerar valencias mayores en los vértices. Esta misma línea se puede seguir en otras superficies.

Con estas consideraciones creemos cumplido el objetivo de crear una idea en el lector sobre la magnitud de esta nueva línea de investigación en superficies, pero quedan sin citar innumerables problemas abiertos en este ámbito.

## Capítulo 4

# Inmersiones ortogonales de grafos en superficies

Este capítulo presenta dos partes distinguibles pero muy relacionadas. En la primera sección presentamos el problema de *la inmersión ortogonal en superficies* y demostramos su naturaleza NP-dura: dado un 4-grafo, obtener una inmersión ortogonal de él que minimize el número de codos entre todas las posibles en cualquier superficie. Este carácter conduce al diseño de algoritmos efectivos, que aunque no proporcionen el número mínimo, construyan inmersiones con un número bajo de codos. En una segunda parte, el algoritmo GPCS, a partir de un 4-grafo 2-conexo con  $n$  vértices, origina una inmersión ortogonal de él en una superficie de género  $c$ , con, a lo más,  $2(l + 1)$  codos en cada arista, donde  $l$  es el número de cruces que presenta dicha arista en un dibujo plano del grafo. El tiempo empleado por este algoritmo es  $O(n + c)$ . Si la entrada la forma una inmersión de un 4-grafo 2-conexo en una superficie  $S$ , el algoritmo IPCS constituye una variante del anterior, dando lugar a otra inmersión, posiblemente distinta, del grafo en una superficie  $S'$ .

## 4.1 Introducción

El problema de minimizar el número de agujeros en una placa, cambios de una cara a otra del circuito, se modeliza en el problema del género de un grafo, este es, determinar la superficie de menor género en la que el grafo dado admite una inmersión. Fue C. Thomassen [73] quien demuestra en 1989 su naturaleza NP-dura. Este resultado hace que desistamos del diseño de una placa con el mínimo número de agujeros.

Otras cantidades se deben minimizar en un circuito, como es el número de codos en los trazados horizontales y verticales entre terminales. Partiendo de un grafo y trabajando en el plano, Garg y Tamassia [31] prueban que comprobar si un grafo admite una inmersión rectilínea plana es un problema NP-completo. Como hemos comentado anteriormente, las placas se diseñan sobre superficies en general no planas, por lo que parece interesante plantearnos el mismo problema en otras superficies. Bajo este incentivo, el problema IRS se enuncia de la siguiente forma: dado un grafo verificar si admite una inmersión rectilínea en alguna superficie. En la Sección 4.2 demostramos que IRS es un problema NP-completo, por lo que su correspondiente problema de optimización es NP-duro.

Para problemas intratables computacionalmente, la línea que se sigue es obtener estructuras, aunque no óptimas, sí "buenas". En contraposición al resultado de Garg y Tamassia [31], diversos autores, como Storer [63], Tamassia y Tollis [70], Kant [40], etc., proporcionan algoritmos polinomiales para construir inmersiones ortogonales planas de grafos dados con un número lineal de codos.

Otra práctica utilizada es la de obtener inmersiones ortogonales con pocos codos en función de los que presente cada arista. En este sentido podemos citar a Even y Granot [25], tres codos por arista, y a Liu, Morgana y Simeone [53], que diseñan un algoritmo polinomial que construye inmersiones ortogonales planas de grafos 2-conexos con, a lo más, dos codos por arista (tres

si es el caso del octaedro).

La Sección 4.3 sigue esta línea. Puesto que la naturaleza del problema no facilita la obtención de una inmersión ortogonal óptima de un grafo dado, entre todas las inmersiones posibles en cualquier superficie, construimos algoritmos lineales para grafos 2-conexos. El algoritmo GPCS parte de un grafo de este tipo y el IPCS de una inmersión.

## 4.2 Minimizar el número de codos es NP-duro

En la línea de obtener inmersiones ortogonales, motivados por el diseño de circuitos impresos, nuestro siguiente objetivo consiste en definir una representación ortogonal de un grafo dado, en alguna superficie de forma que minimize el número de codos. Este capítulo está dedicado a probar la naturaleza NP-dura de este problema.

En el lenguaje de la NP-completitud la cuestión planteada correspondería a un problema de búsqueda y si hablamos de inmersiones rectilíneas estaríamos considerando su correspondiente problema de decisión, cuya naturaleza NP-completa nos conduciría a la NP-dureza del de optimización. En este contexto enunciamos el problema de decisión de la siguiente forma:

**INMERSIÓN RECTILÍNEA EN SUPERFICIES (IRS):**

**ENTRADA:** Grafo  $G$ .

**PREGUNTA:** ¿Existe una superficie  $S$  y una inmersión rectilínea de  $G$  en  $S$ ?

En la parte del Capítulo 0 dedicada a los temas de NP-completitud estudiamos dos problemas NP-completos estándares, 3-SAT y NE3SAT. El primero de ellos fue considerado en el Capítulo 3 y es el segundo el que nos va a permitir obtener nuestro resultado.

**Teorema 4.2.1.** *El problema de la inmersión rectilínea en superficies es de naturaleza NP-completa.*

**Demostración:** La prueba está compuesta de dos partes: en la primera probaremos que IRS pertenece a la clase de problemas NP y en la segunda definiremos una transformación polinomial de NE3SAT en IRS ( $\text{NE3SAT} \in \text{IRS}$ ).

Partimos de un grafo  $G$  como entrada del problema. Para el primer punto consideramos un algoritmo no determinista que, una vez generada una inmersión de  $G$  en una cierta superficie, en la etapa “checking”, comprueba si dicha inmersión es rectilínea. Como se probó en el Teorema 3.2.1 esta operación se realiza en tiempo polinomial en el número de aristas de  $G$ , luego es posible diseñar un algoritmo no determinista polinomial que comprueba la veracidad de IRS.

En una segunda etapa demostramos que  $\text{NE3SAT} \in \text{IRS}$ ; para ello definiremos en primer lugar una aplicación polinomial que asocia a cada entrada de NE3SAT otra de IRS y posteriormente probamos la equivalencia entre sus respuestas afirmativas.

Sea  $U = \{u_1, u_2, \dots, u_n\}$  un conjunto de variables y  $C = \{c_1, c_2, \dots, c_m\}$  un conjunto de cláusulas sobre  $U$ , formando una entrada arbitraria  $\mathcal{R}$  de NE3SAT. Partiendo de  $\mathcal{R}$  construimos un grafo  $G_{\mathcal{R}}$  de la siguiente forma: a cada cláusula, excepto a las que contienen a un literal y a su negación, se le asocia un vértice, y cada variable  $u_i$ ,  $i = 1, \dots, n$  se representa por una secuencia de ciclos  $C_3$  de longitud 3, adyacentes al ciclo anterior y posterior mediante vértices distintos. Los ciclos de la secuencia en posición impar representarán a la variable en su forma afirmativa y aquellos que ocupen posición par estarán ligados a la variable en su forma negativa. La longitud de estas secuencias vendrá determinada por el número de veces que aparece la variable en las cláusulas, en cualquiera de sus dos formas, afirmativa o negativa, de tal manera que existan suficientes vértices de valencia dos en la cadena para poder unirlos con los vértices que representan las cláusulas, aumentando de

esta forma la valencia de aquellos vértices al pasar de grado dos a tres, esto es, cada vértice que representa a una cláusula tendrá valencia tres, una arista por cada literal, incidente en uno de los vértices de valencia dos perteneciente a uno de los ciclos asociados a dicho literal. De esta forma obtenemos un grafo  $G_{\mathcal{R}}$  con vértices de valencia tres representando a las cláusulas y secuencias de ciclos  $C_3$  representando a los literales, con vértices de valencia dos y tres, estos últimos correspondientes a adyacencias entre ciclos o con cláusulas.

Para definir las adyacencias entre literales y cláusulas se sigue el orden numérico de éstas y en cuanto a los ciclos, sus vértices se van tomando siguiendo la secuencia. En un último paso se eliminan los ciclos cuyas únicas adyacencias sean con vértices de otros ciclos que también estén en esta situación.

La Figura 4.1 muestra el grafo construido a partir del conjunto de variables  $U = \{u_1, u_2, u_3, u_4\}$  y del conjunto de cláusulas  $C = \{c_1, c_2, c_3\}$  donde  $c_1 = \{u_1, \bar{u}_2, u_4\}$ ,  $c_2 = \{\bar{u}_1, u_3, u_4\}$  y  $c_3 = \{u_2, u_3, u_4\}$ .

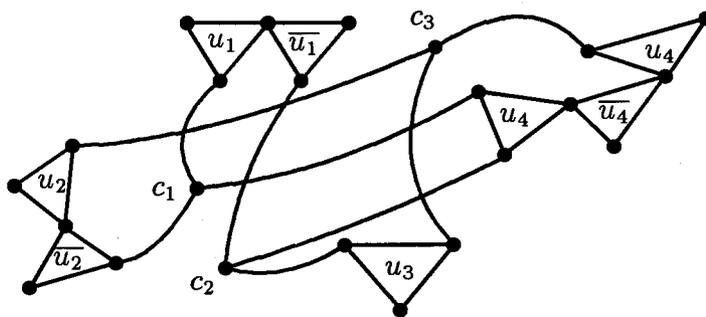


Figura 4.1: Un ejemplo para tres cláusulas.

Para construir la secuencia de ciclos asociados a una variable, a lo más, consideramos  $2m$  ciclos, luego en total necesitamos  $O(nm)$ . El grafo  $G_{\mathcal{R}}$  contiene, como máximo,  $m$  vértices asociados a las cláusulas, cada uno de ellos de valencia 3. Estas observaciones nos llevan a asegurar que la transformación

definida es polinomial en el número de variables y cláusulas.

Para concluir demostramos la equivalencia entre las respuestas afirmativas a las entradas  $\mathcal{R}$  y  $G_{\mathcal{R}}$ . Trabajamos, como hemos comentamos en varias ocasiones, con las representaciones ortogonales estándares de las superficies, por lo que la diferencia entre unas y otras será el número de asas o ventanas, característica que no influye en la rectilínealidad; por tanto podemos tratar indistintamente a inmersiones rectilíneas de  $G_{\mathcal{R}}$  en superficies distintas.

En primer lugar analizando las características de cualquier inmersión rectilínea de  $G_{\mathcal{R}}$  llegamos a los siguientes hechos:

- a) La longitud de los ciclos de las secuencias, longitud tres, da lugar a una única inmersión rectilínea posible de  $C_3$ , ésta es, una circunferencia horizontal o vertical, en la que cada arista forma un ángulo de 180 grados con la siguiente en el ciclo. Por lo tanto, las tres aristas de un mismo ciclo presentarán la misma dirección, horizontal o vertical, en la circunferencia asociada a éste, pudiendo hablar, por tanto, de la dirección del ciclo en la inmersión rectilínea.
- b) Las inmersiones rectilíneas de dos ciclos asociados a la misma variable presentan la misma dirección si y sólo si la paridad de la posición que ocupan en la secuencia es la misma.
- c) Las inmersiones rectilíneas de dos ciclos asociados a la misma variable presentan direcciones distintas si y sólo si sus posiciones en la secuencia tienen distinta paridad.

Como conclusión de las aseveraciones anteriores, dada una inmersión rectilínea de  $G_{\mathcal{R}}$ , obtenemos una correspondencia entre los literales asociados a cada variable y las dos direcciones vertical y horizontal; a cada literal  $u_l$  ( $\bar{u}_l$ ) le hacemos corresponder la dirección  $d_l$  ( $\bar{d}_l$ ) que presentan sus ciclos asociados.

Este hecho conduce a la siguiente asignación de verdad sobre  $U$ :

$$t(u_i) = \begin{cases} T & \text{si } d_i \text{ es vertical} \\ F & \text{si } d_i \text{ es horizontal} \end{cases}$$

donde  $T$  corresponde al valor *verdadero* (dirección vertical) y  $F$  al valor *falso* (dirección horizontal).

Por otro lado, como cada vértice asociado a una cláusula tiene valencia tres, entre sus aristas incidentes aparecerá una con dirección vertical y otra con dirección horizontal. La arista vertical será incidente con un vértice de un ciclo asociado al literal correspondiente, y necesariamente dicho ciclo tiene que presentar dirección horizontal, es decir el literal tiene asignado, según la aplicación anterior, el valor falso; un razonamiento análogo se seguiría para la arista horizontal, incidente en un vértice asociado a un literal con valor verdadero.

Estas aseveraciones demuestran que la aplicación sobre  $U$  definida anteriormente hace que todas las cláusulas contengan un literal verdadero y otro falso. Las cláusulas que contienen a un literal y a su negación, no representadas en  $G_{\mathcal{R}}$  se satisfacen trivialmente y a las variables que no estén representadas en ninguna cláusula se le asigna un valor arbitrario. Por tanto, a partir de una inmersión rectilínea de  $G_{\mathcal{R}}$  en una cierta superficie, hemos definido una asignación de verdad satisfactoria para  $\mathcal{R}$ .

Para finalizar la prueba vamos a construir una inmersión rectilínea de  $G_{\mathcal{R}}$  en una cierta superficie a partir de una asignación de verdad satisfactoria para  $\mathcal{R}$ .

Comenzamos considerando el toro para después aumentar su género en la medida que lo necesitemos. Para cada ciclo asociado a un literal verdadero elegimos como circunferencia un meridiano y para los asociados a literales falsos un paralelo, teniendo en cuenta que en este punto no se sitúan, aún, sus vértices. Los puntos de la superficie que representan a los vértices correspondientes a las cláusulas  $c_i$ ,  $i = 1, \dots, m$ , son elegidos arbitrariamente sobre

ella sin que estén contenidos sobre las circunferencias (paralelos y meridianos) elegidas. Las aristas que parten de  $c_i$  y que se corresponden con literales verdaderos son incidentes a meridianos, por lo que su dirección tiene que ser horizontal; lo contrario ocurre con los falsos. Esto nos conduce a dibujar las aristas “verdaderas” horizontales y las “falsas” verticales, siguiendo el mismo criterio de asignación que en el paso anterior. Como cada  $c_i$  tiene valencia 3 y hemos partido de una asignación de verdad satisfactoria, en cada uno de estos vértices hay que dibujar dos aristas verticales y una horizontal o viceversa. El sentido de dichas aristas queda determinado imponiendo la condición de que si al trazar una de ellas, partiendo de  $c_i$ , tenemos que cruzar una circunferencia a la cual también es incidente  $c_i$ , entonces elegimos el sentido contrario. El trazado de estas aristas determina puntos sobre las circunferencias, lugares que elegimos para situar los vértices correspondientes. Los restantes vértices de valencia dos de los ciclos se sitúan aleatoriamente sobre las circunferencias asociadas.

Esta inmersión es considerada sobre una superficie de género una unidad más del cuadrado del número de aristas de  $G_{\mathcal{R}}$ , para así evitar cruces inapropiados entre las aristas.

Queda así demostrada la equivalencia entre las respuestas afirmativas de ambos problemas, por lo que concluimos con la naturaleza NP-completa del problema IRS. ■

La Figura 4.2 muestra una inmersión rectilínea del grafo representado en la Figura 4.1, sobre una superficie de género 16, obtenida a partir de la siguiente asignación de verdad satisfactoria para las cláusulas dadas:

$$\begin{aligned} t(u_1) = 1 & \quad t(u_3) = 0 \\ t(u_2) = 1 & \quad t(u_4) = 1 \end{aligned}$$

De forma inmediata se obtiene la naturaleza NP-dura del correspondiente problema de optimización, que enunciamos como sigue:

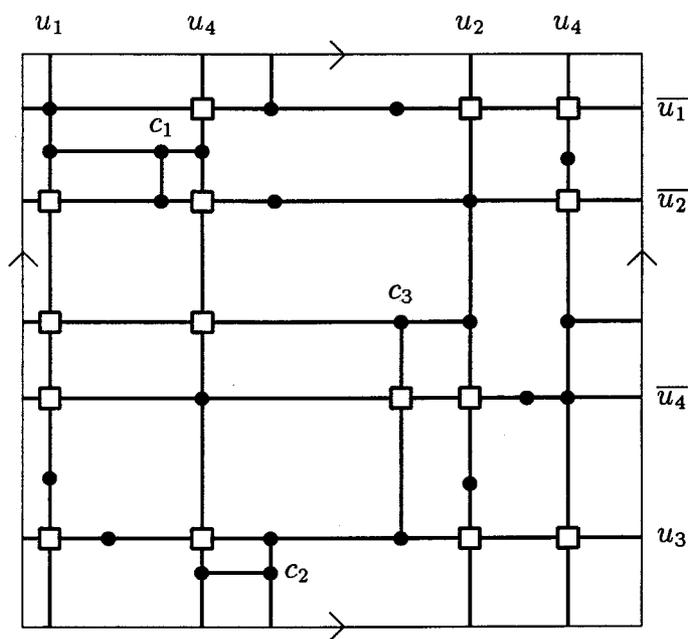


Figura 4.2: Inmersión rectilínea del grafo de la Figura 4.1.

**Corolario 4.2.2.** *Dado un grafo  $G$ , encontrar una superficie  $S$  y una inmersión ortogonal  $\phi$  de  $G$  en  $S$  tal que  $\phi$  minimize el número de codos entre todas las inmersiones de  $G$  en cualquier superficie es un problema NP-duro.*

### 4.3 Inmersiones ortogonales con pocos codos

La naturaleza NP-dura del problema planteado en la sección anterior hace que, en lugar de buscar el mínimo número de codos, nos planteemos el diseño de algoritmos polinomiales que proporcionen inmersiones ortogonales de grafos en superficies con un número pequeño de codos. Este recurso, como apuntábamos anteriormente, es muy utilizado en el diseño de circuitos impresos con propiedades que originan problemas de la misma naturaleza.

Siguiendo esta línea presentamos un algoritmo lineal que, partiendo de un 4-grafo 2-conexo de  $n$  vértices, en general no plano, proporciona una inmersión ortogonal de él en una superficie, con, a lo más,  $2(l + 1)$  codos en cada arista, donde  $l$  es el número de cruces que ésta presenta en un dibujo plano del grafo. Este algoritmo es *output sensitive*, es decir su complejidad depende de los datos de salida, en este caso  $O(n + c)$  donde  $c$  será el género de la superficie. No abordamos los detalles técnicos del algoritmo, sino que exponemos las operaciones generales que hay que seguir en cada paso.

El algoritmo recibe como entrada en el **Paso 1** un 4-grafo  $G$  2-conexo.

El **Paso 2** proporciona un dibujo de  $G$  en el plano. Puesto que en general  $G$  no va a ser plano, no será posible obtener una inmersión plana de él, luego el dibujo presentará cruces entre aristas en puntos que no representan vértices. Imponemos que en estos puntos se produzca un único cruce entre dos aristas (carencia de cruces múltiples), propiedad que podemos obtener modificando ligeramente la posición de las aristas. Para no trabajar con un número elevado de estos puntos, la situación óptima sería construir un dibujo con el mínimo número de cruces entre ellas. Sin embargo este problema es

el llamado *número de cruces* (crossing number) de un grafo y como sabemos Garey y Johnson [30] probaron en 1983 su naturaleza NP-dura. Como situación extrema, cada arista puede estar cruzada, a lo más, por todas las restantes, luego  $\frac{n(n-1)}{2} - 1$  cruces en cada arista en el peor de los casos.

Denotemos por  $c - 1$  al número de cruces totales producidos para un grafo con  $n$  vértices y  $m$  aristas. Si consideramos este dibujo inmerso en el toro y colocamos una ventana en cada uno de los cruces, podemos suponer que estamos trabajando con una inmersión de  $G$  en una superficie de género  $c$ , de forma que ninguna arista intersecta al rectángulo exterior de su representación ortogonal estándar. En realidad estamos haciendo uso del homeomorfismo existente entre el plano con  $c - 1$  ventanas y la superficie de género  $c$  a la que se le han eliminado dos circunferencias adyacentes por un punto (lo que se denomina un  $\delta$ ).

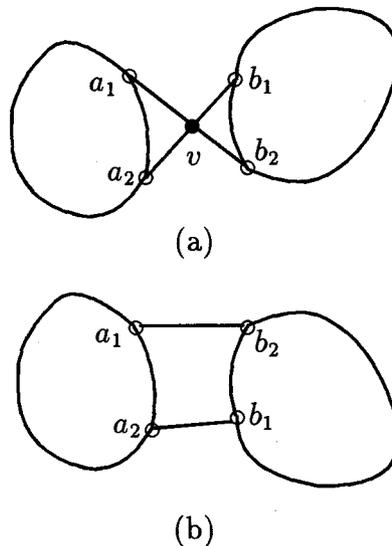


Figura 4.3: (a) El vértice  $v$  es un vértice de corte. (b) El *shift* realizado elimina el cruce o vértice virtual.

En estas condiciones el **Paso 3** elimina los cruces o ventanas situando

un vértice en cada uno de ellos, llamémosles vértices virtuales, los cuales tendrán valencia cuatro debido a la unicidad de los cruces en los puntos del plano.

En general, el grafo plano así obtenido no será 2-conexo, localizándose los posibles puntos de corte en vértices virtuales (debido a la 2-conexión de  $G$ ). Si uno de estos vértices es punto de corte es posible obtener otro dibujo en el que se ha eliminado el cruce correspondiente realizando un *shift* en el dibujo, tal como indica la Figura 4.3.

El **Paso 4** realiza un *shift* en el dibujo por cada vértice de corte.

Por tanto, podemos suponer que el grafo plano  $G'$  de  $n + c - 1$  vértices y  $m + 2(c - 1)$  aristas es 2-conexo de valencia menor o igual que cuatro. Estas propiedades nos indican que es posible aplicar el algoritmo presentado en [53] a  $G'$ , operación que se realiza en el **Paso 5**, obteniéndose una inmersión ortogonal plana de  $G'$  con, a lo más, 2 codos por arista (tres si el grafo de partida es el octaedro).

El **Paso 6** consiste en eliminar los vértices virtuales añadidos y situar una ventana en cada cruce, obteniendo de esta forma una inmersión ortogonal de  $G$  en una superficie de género  $c$ . El número de codos que presenta cada arista de  $G$  viene dado en función del número de cruces que presentaba en su dibujo plano original. Se obtiene el caso óptimo cuando el número de cruces del dibujo considerado sea justamente el número de cruces de un grafo, que para el grafo completo  $K_n$ ,  $O(n^4)$  es una cota superior de él [34]; en el peor de los casos cada arista estaría cruzada por todas las demás. Denotamos por  $l$  al número de cruces que presenta una arista en el dibujo plano de partida. En definitiva concluimos que la inmersión ortogonal de  $G$  en la superficie de género  $c$  presenta, como máximo,  $2(l + 1)$  codos en cada arista.

A continuación esquematizamos el algoritmo de la siguiente forma:

Algoritmo GPCS

**Entrada:** 4-Grafo  $G$  de  $n$  vértices,  $m$  aristas y 2-conexo.

**Salida:** Superficie  $S$  e inmersión ortogonal de  $G$  en  $S$  con, a lo más,  $2(l + 1)$  codos en cada arista, donde  $l$  es el número de cruces que presenta la arista en un dibujo plano de  $G$ .

**Paso 1.** El algoritmo recibe como entrada al grafo  $G$ .

**Paso 2.** Se construye un dibujo plano de  $G$  sin cruces múltiples entre las aristas.

**Paso 3.** Se sitúa un vértice virtual en cada cruce.

**Paso 4.** Para cada vértice de corte se realiza un shift.

**Paso 5.** Se aplica el algoritmo de [53] al grafo obtenido.

**Paso 6.** Se eliminan los vértices virtuales y se coloca una ventana en cada cruce resultante.

**Paso 7.** FIN

Pasamos al estudio asintótico del algoritmo GPCS. Los Pasos 1 y 2 se realizan en tiempo lineal en el número de vértices de  $G$ . Para situar los vértices virtuales se emplea tiempo  $O(c)$  y una vez considerados se localizan los vértices de corte y se realizan los shifts en tiempo lineal en función del número de vértices totales del grafo  $G'$ , es decir  $O(n + c)$ . Esta última es también la complejidad del Paso 5 para obtener la inmersión ortogonal en virtud del algoritmo de [53]. Por último la eliminación de los vértices virtuales y la consecución de la superficie de género  $c$  vuelve a requerir tiempo  $O(c)$ . Queda entonces probado el siguiente teorema:

**Teorema 4.3.1.** *El algoritmo GPCS proporciona en tiempo  $O(n + c)$  una inmersión ortogonal de un 4-grafo 2-conexo de  $n$  vértices, en una superficie de género  $c$ , con, a lo más,  $2(l + 1)$  codos en cada arista, donde  $l$  es el número de cruces que presenta la arista en un dibujo plano del grafo.*

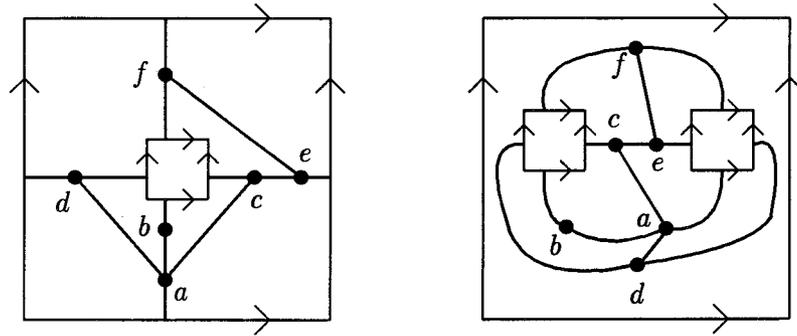


Figura 4.4: Aumentando en una unidad el género de la superficie, se obtiene una inmersión cuyas aristas no intersectan al rectángulo exterior.

Como hemos indicado, la entrada del algoritmo GPCS está formada por un 4-grafo; podemos dar una variante, algoritmo IPCS, en la que la entrada estaría compuesta por una inmersión de un grafo en una superficie de género  $a$  y la salida sería una inmersión ortogonal de dicho grafo, generalmente no equivalente a la dada, en una superficie de género  $c$ . Partiendo de una inmersión en una superficie, el algoritmo que vamos a detallar requiere que ninguna de sus aristas intersecte al rectángulo exterior de su representación ortogonal estándar. Para ello, el segundo paso del algoritmo modifica la inmersión para llevarla a una con dicha característica. La idea de la transformación queda explícita en la Figura 4.4. El proceso consiste en aplicar el homeomorfismo existente entre la superficie de partida a la que se le ha eliminado un punto y la superficie de género una unidad más.

Explicitamos el algoritmo de la siguiente forma:

Algoritmo IPCS

**Entrada:** Inmersión  $\phi$  de un 4-grafo  $G$  2-conexo de  $n$  vértices y  $m$  aristas en una superficie  $S$ .

**Salida:** Inmersión ortogonal  $\phi'$  de  $G$  en una superficie  $S'$  con, a lo más,  $2(l+1)$  codos en cada arista, donde  $l$  es el número de cruces de ésta si  $\phi$  se considera un dibujo plano.

**Paso 1.** El algoritmo recibe como entrada a la inmersión  $\phi$ .

**Paso 2.** En caso de que alguna arista de  $\phi$  intersecte al rectángulo exterior de la representación ortogonal estándar de la superficie, se aumenta en una unidad el género de la superficie para que ello no ocurra (ver Figura 4.4).

**Paso 3.** Se eliminan las ventanas y se sitúan vértices virtuales distintos en cada uno de los cruces resultantes. Si alguno de éstos es múltiple, se modifica ligeramente la posición de las aristas para obtener vértices virtuales de valencia cuatro.

**Paso 4.** Para cada vértice virtual de corte se realiza un shift.

**Paso 5.** Se aplica el algoritmo expuesto en [53] al grafo obtenido.

**Paso 6.** Se eliminan los vértices virtuales de la inmersión ortogonal  $\phi'$  obtenida y se colocan ventanas para evitar los cruces.

**Paso 7.** FIN

Notemos que la posible realización de shifts, hace que la inmersión obtenida pueda no ser equivalente a la dada, tal y como muestra la Figura 4.5.

No podemos establecer relación entre los géneros de la superficie de entrada  $S$  y la de salida  $S'$ . Pueden existir ventanas en  $S$  que no correspondan a cruces inapropiados entre aristas, en las que no hay que situar vértices virtuales; en este sentido  $a$  puede ser mayor que  $c$ . Sin embargo si aparecen

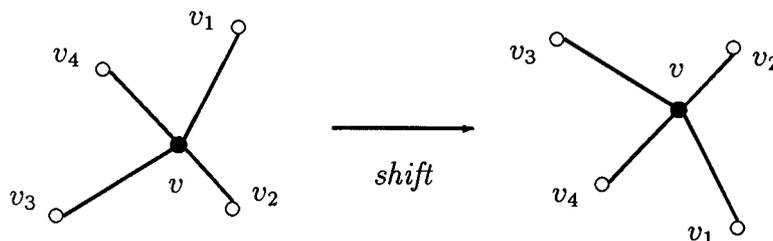


Figura 4.5: La rotación en el vértice  $v$  cambia al realizar un shift.

asas en  $S$  con las que se evitan más de un cruce, hay que situar más de un vértice virtual, luego el género de  $S'$  será mayor que el de  $S$ .

Como conclusión obtenemos el siguiente resultado:

**Teorema 4.3.2.** *El algoritmo IPCS proporciona, a partir de una inmersión en una superficie  $S$  dada y en tiempo  $O(n + c)$ , una inmersión ortogonal de un 4-grafo 2-conexo de  $n$  vértices en una superficie  $S'$  de género  $c$ , con, a lo más,  $2(l + 1)$  codos en cada arista, donde  $l$  es el número de cruces que ésta presenta si la inmersión de partida se considera plana.*

## 4.4 Conclusiones y problemas abiertos

Se ha presentado esta parte como el inicio de una nueva línea de investigación, en la que se abordan problemas sobre inmersiones ortogonales de grafos en distintas superficies. El incentivo de esta vía es el diseño de circuitos impresos, los cuales se realizan sobre placas que determinan superficies arbitrarias, ya que en general, el grafo ha dibujar no es plano.

Nos centramos en el problema de minimizar el número de codos que

presenta un circuito, modelizando la situación bajo el estudio de inmersiones ortogonales óptimas.

Bajo esta perspectiva, nos hemos cuestionado la existencia de una inmersión rectilínea de un grafo dado entre todas las representaciones que admita en todas las superficies; el Teorema 4.2.1 demuestra la naturaleza NP-completa del problema. De forma inmediata se concluye que el correspondiente problema de optimización es NP-duro.

Ante esta naturaleza del problema, proporcionamos aproximaciones de la solución óptima. Los algoritmos GPCS y IPCS construyen inmersiones ortogonales en una superficie de género  $c$  y en tiempo  $O(n + c)$ , partiendo de un grafo o de otra inmersión, respectivamente, ambos 2-conexos y con  $n$  vértices, que presentan, a lo más,  $2(l + 1)$  codos en cada arista, donde  $l$  es el número de cruces originados en la arista si se consideran dibujos planos. Puesto que, en general, estas cantidades no son óptimas, se abre la posibilidad de mejorar estas cotas sobre el número de codos.

Análogamente a lo que ocurría en el capítulo anterior, se presentan muchos y muy diversos problemas abiertos. Garg y Tamassia [31] han demostrado que obtener una inmersión rectilínea plana de un 4-grafo dado, es un problema NP-completo. Nosotros hemos querido obtener una inmersión rectilínea en superficies cualesquiera, obteniendo el mismo resultado. Planteando la cuestión en una superficie fija no plana, surgen distintos problemas aún no estudiados. Lo que sí es claro es que el grafo tiene que admitir una inmersión en la superficie dada, para poder hablar de inmersiones ortogonales. En esta línea el primer intento, motivado por los antecedentes, es demostrar también una naturaleza intratable, aunque no se descarta la posibilidad de poder obtener algoritmos eficientes.

# Referencias

- [1] A. V. Aho, J. E. Hopcroft y J. D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.
- [2] P. Bachmann. *Die analytische Zahlentheorie*. Teubner, Leipzig, 1894.
- [3] C. Batini, M. Talamo y R. Tamassia. An algorithm for automatic layout of entity relationship diagrams. En P. A. Ng C. G. Davis, S. Jajodia y R. T. Yeh, editores, *Entity-Relationship Approach to Software Engineering, Proc. 3rd Internat. Conf. on Entity-Relationship Approach*, páginas 421–439. North-Holland, Amsterdam, Nueva York, 1983.
- [4] C. Batini, M. Talamo y R. Tamassia. Computer aided layout of entity-relationship diagrams. *J. Systems Software*, 4:163–173, 1984.
- [5] G. Di Battista, P. Eades, H. de Fraysseix, R. Tamassia e I. G. Tollis, editores. *Graph Drawing'93, Proceedings of the ALCOM International Workshop on Graph Drawing*, Paris, Francia, Septiembre 1993.
- [6] G. Di Battista, P. Eades, R. Tamassia e I. G. Tollis. Algorithms for drawing graphs: an annotated bibliography. *Computational Geometry: Theory and Applications*, 4(5):235–282, 1994.
- [7] G. Di Battista, G. Liotta y F. Vargiu. Spirality of orthogonal representations and optimal drawings of series-parallel graphs and 3-planar graphs. En R. Tamassia e I. G. Tollis, editores, *Proc. Workshop on Algorithms*

- and Data Structures, *Lectures Notes in Computer Science*, 709, páginas 151–162. Springer-Verlag, 1993.
- [8] S. Bhatt y S. Cosmadakis. The complexity of minimizing wire lengths in VLSI layouts. *Information Processing Letters*, 25:263–367, 1987.
- [9] T. Biedl. Orthogonal graphs drawings. Diplomarbeit, Fachbereich Mathematik, Technische Univ., 1995.
- [10] T. Biedl. Improved orthogonal drawings of 3-graphs. Rutcor Research Report 15-96, 1996, disponible en <http://rutcor.rutgers.edu/rrr>.
- [11] T. Biedl. Optimal orthogonal drawings of connected plane graphs. Rutcor Research Report 11-96, 1996, disponible en <http://rutcor.rutgers.edu/rrr>.
- [12] T. Biedl y G. Kant. A better heuristic for orthogonal graph drawings. En *Proc. of the 2nd European Symp. on Algorithms (ESA 94)*, *Lecture Notes in Comp. Science* 855, páginas 124–135. Springer-Verlag, 1994.
- [13] F. J. Brandenburg. Nice drawings of graphs and trees are computationally hard. Technical Report MIP-8820, 1988.
- [14] F. J. Brandenburg, editor. *Symposium on Graph Drawing*, GD 95, volumen 1027 de *Lecture Notes in Computer Science*, Passau, Alemania, Septiembre 1995. Springer Verlag.
- [15] A. L. Cauchy. Recherches sur les polyédres-premier mémoire. *Journal de l'École Polytechnique*, 9(Cah. 16):68–86, 1813.
- [16] G. Chartrand y L. Lesniak. *Graphs and Digraphs*. Wadsworth and Brooks/Cole, 1986.
- [17] N. Chiba, T. Nishizeki, S. Abe y T. Ozawa. A linear algorithm for embedding planar graphs using PQ-trees. *J. Comput. Syst. Sci.*, 30(1):54–76, 1985.

- [18] A. Cobham. The intrinsic computational difficulty of functions. En Bar-Hillel, editor, *Proc. 1964 International Congress for Logic Methodology and Philosophy of Science*, páginas 24–30. North Holland, Amsterdam, 1964.
- [19] F. J. Cobos-Gavala. *Sobre visibilidad en espacios  $n$ -dimensionales y superficies no planas*. Tesis Doctoral, Dpto. de Matemática Aplicada I, Universidad de Sevilla, 1995.
- [20] S. A. Cook. The complexity of theorem-proving procedures. En Association for Computing Machinery, editor, *Proc. 3rd Ann. ACM Symp. on Theory of Computing*, páginas 151–158. Nueva York, 1971.
- [21] D. Dolev, F. T. Leighton y H. Trickey. Planar embeddings of planar graphs. En F. P. Preparata, editor, *Advances in Computer Research*, volume 2: VLSI Theory, páginas 147–161, Greenwich, CT, 1984. JAI Press Inc.
- [22] J. Edmonds. Paths, trees and flowers. *Canad. J. Math.*, 17:449–467, 1965.
- [23] J. R. Edmonds. A combinatorial representation for polyhedral surfaces. *Notices Amer. Math. Soc.*, 7:646, 1960.
- [24] S. Even y G. Granot. Grid layouts of block diagrams—bounding the number of bends in each connection. En R. Tamassia e I. G. Tollis, editores, *Graph Drawing 94, Lectures Notes in Computer Science*, páginas 64–75. Springer-Verlag, 1994.
- [25] S. Even y G. Granot. Rectilinear planar drawings with few bends in each edge. Tech. Report 797, Comp. Science Dept., 1994.
- [26] S. Even y R. E. Tarjan. Computing an st-numbering. *Theor. Comput. Sci.*, 2:339–344, 1976.
- [27] M. Formann y F. Wagner. The VLSI layout problem in various embedding models. En *Graph-Theoretic Concepts in Computer Science (16th*

- Workshop WG'90*), páginas 130–139, Berlin/Heidelberg, 1992. Springer Verlag.
- [28] M. Fréchet y Ky Fan. *Initiation to Combinatorial Topology*. Weber and Schmidt, 1967.
- [29] M. R. Garey y D. S. Johnson. *Computers and Intractability: a guide to the theory of NP-completeness*. Freeman, 1979.
- [30] M. R. Garey y D. S. Johnson. Crossing number is NP-complete. *SIAM J. Algebraic Discrete Methods*, 4(3):312–316, 1983.
- [31] A. Garg y R. Tamassia. On the computational complexity of upward and rectilinear planarity testing. En R. Tamassia e I. G. Tollis, editores, *Graph Drawing 94, Lectures Notes in Computer Science*, páginas 286–297. Springer-Verlag, 1994.
- [32] A. Gregori. Unit length embedding of binary trees on a square grid. *Information Processing Letters*, 31:167–172, 1989.
- [33] J. L. Gross y T. W. Tucker. *Topological graph theory*. John Wiley & Sons, 1987.
- [34] R. K. Guy. The decline and fall of zarankiewicz's theorem. En F. Harary, editor, *Proof Techniques in Graph Theory*. Academic Press, Nueva York, 1969.
- [35] F. Harary. *Graph Theory*. Addison Wesley Publishing Company, 1972.
- [36] L. Heffter. Über das Problem der Nachbargebiete. *Math. Ann.*, 38:477–508, 1891.
- [37] J. E. Hopcroft y R. E. Tarjan. Efficient planarity testing. *J. Assoc. Comput. Mach.*, 21:549–568, 1974.
- [38] P. J. Idicula. Drawing trees in grids. Master's thesis, Department of Computer Science, University of Auckland, 1990.

- [39] A. P. Juškevič y E. Winter. *Leonhard Euler und Christian Goldbach: Briefwechsel 1729-1764*. Akademie-Verlag, Berlin, 1965.
- [40] Goos Kant. Drawing planar graphs using the lmc-ordering. En *Proc. 33th Ann. IEEE Symp. on Found. of Comp. Science*, páginas 101–110, 1992.
- [41] R. M. Karp. Reducibility among combinatorial problems. En R. E. Miller y J. W. Thatcher, editores, *Complexity of Computer Computations*, páginas 85–103. Plenum Press, Nueva York, 1972.
- [42] D. E. Knuth. Big omicron and big omega and big theta. *SIGACT News*, 8(2):18–24, 1976.
- [43] M. R. Kramer y J. van Leeuwen. The complexity of wire routing and finding minimum area layouts for arbitrary VLSI circuits. En F. P. Preparata, editor, *Advances in Computer Research*, volume 2: VLSI Theory, páginas 129–146. JAI Press, 1992.
- [44] K. Kuratowski. Sur le problème des courbes gauches en topologie. *Fund. Math.*, 15:271–283, 1930.
- [45] E. L. Lawler. *Combinatorial Optimizacion: Networks and Matroids*. Holt, Rinehart and Winston, Nueva York, 1976. Cap' itulo 4.
- [46] C. E. Leiserson. *Area efficient VLSI computation*. Tesis Doctoral, Carnegie Mellon University, Pittsburgh, 1981. MIT Press, Cambridge, MA, 1983.
- [47] T. Lengauer. *Combinatorial Algorithms for integrated circuit layout*. Teubner/Wiley & Sons, Stuttgart/Chicester, 1990.
- [48] S. A. J. Lhuillier. Mémoire sur la polyédrométrie. *Annales de Mathématiques*, 3(3):169–189, 1812.
- [49] R. Liggett y W. Mitchell. Optimal space planning in practice. *Computer Aided Design*, 13:277–288, 1981.

- [50] Y. Liu, P. Marchioro y R. Petreschi. At most single-bend embeddings of cubic graphs. por aparecer, 1994.
- [51] Y. Liu, P. Marchioro, R. Petreschi y B. Simeone. Theoretical results on at most 1-bend embeddability of graphs. Technical Report, Dept. of Statistics, 1990.
- [52] Y. Liu, A. Morgana y B. Simeone. General theoretical results on rectilinear embeddability of graphs. *Acta Math. Appl. Sinica*, 7:187–192, 1991.
- [53] Y. Liu, A. Morgana y B. Simeone. A linear time algorithm for finding 3-embeddings of a graph. Research Report, 1993.
- [54] E. F. Moore. The shortest path through a maze. En Harvard Univ. Press, editor, *Proc. Internat. Symp. Switching Theory*, páginas 285–292. Cambridge, 1959.
- [55] T. Nishizeki y N. Chiba. *Planar Graphs: Theory and Algorithms*. North-Holland Mathematics Studies, 1988.
- [56] Stephen North, editor. *Symposium on Graph Drawing, GD 96*, Lecture Notes in Computer Science, Berkeley, California, Septiembre 1996. Springer Verlag.
- [57] A. Paoluzzi. Il Progetto di Componenti Edilizi Industrializzati ed il Controllo di Qualità attraverso l'Approccio Sistemistico. Technical Report 14, 1978.
- [58] A. Rosenberg. Three-Dimensional VLSI: a case study. *J. ACM*, 30(3):397–416, 1983.
- [59] G. Ruhe. *Algorithmic aspects of flows in networks*. Kluwer, 1991.
- [60] T. J. Schaefer. The complexity of satisfiability problems. En *Proc. 10th Ann. ACM Symp. on Theory of Computing, Association for Computing Machinery*, páginas 216–226, Nueva York, 1978.

- [61] Y. Shiloach. *Arrangements of Planar Graphs on the Planar Lattice*. Tesis Doctoral, Weizmann Institute of Science, Rehovot, Israel, 1976.
- [62] A. J. Storer. The node cost measure for embedding graphs in the planar grid. En *Proc. 12th ACM Symposium on the Theory of Computing*, páginas 201–210, 1980.
- [63] J. Storer. On minimal node-cost planar embeddings. *Networks*, 14:181–212, 1984.
- [64] R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *Siam J. Comput.*, 16(3):421–444, 1987.
- [65] R. Tamassia. Comunicación personal, Diciembre 1995.
- [66] R. Tamassia e I. G. Tollis, editores. *DIMACS International Workshop, Graph Drawing 94*, volume 894 of *Lecture Notes in Computer Science*, Princeton, New Jersey, USA, Octubre 1994. Springer Verlag.
- [67] R. Tamassia e I. G. Tollis. Efficient embedding of planar graphs in linear time. En *Proc. IEEE Int. Symp. on Circuits and Systems*, páginas 495–498. Philadelphia, 1987.
- [68] R. Tamassia e I. G. Tollis. A unified approach to visibility representations of planar graphs. *Discrete Comput. Geom.*, 1:321–341, 1986.
- [69] R. Tamassia e I. G. Tollis. Tessellation representations of planar graphs. *Proc. 27th 1989 Annual Allerton Conference*, páginas 48–57, 1989.
- [70] R. Tamassia e I. G. Tollis. Planar grid embeddings in linear time. *IEEE Trans. on Circuits and Systems*, CAS-36:1230–1234, 1989.
- [71] R. Tamassia e I. G. Tollis. Representations of graphs on a cylinder. *SIAM J. Discrete Mathematics*, 4(1):139–149, 1991.
- [72] R. E. Tarjan. “Data Structures and Network Algorithms”. CBMS-NSF Regional Conference Series in Applied Mathematics, vol.44, Society for Industrial Applied Mathematics, 1983.

- 
- [73] C. Thomassen. The graph genus problem is NP-complete. *Journal of Algorithms*, 10:568–576, 1989.
- [74] C. Thomassen. Embeddings and minors. En R. L. Graham, M. Grotscel, y L. Lovasz, editores, *Handbook of Combinatorics*, páginas 301–350. Elsevier, 1995.
- [75] L. G. Valiant. Universality considerations in VLSI circuits. *IEEE Trans. Comput.*, C-30:135–140, 1981.
- [76] A. T. White. *Graphs, Groups and Surfaces*. North-Holland, Amsterdam, 1973.
- [77] S. K. Wismath. Characterizing bar line-of-sights graphs. *In Proceedings of the 1st Symposium on Computational Geometry*, páginas 147–152, 1985.
- [78] Liu Yanpei. *Embeddability in Graphs*. Kluwer Academic Publishers, 1995.

# Índice

- 3-satisfacibilidad, 29, 144
- algoritmo, 20
  - de aumentación, 42
  - determinista, 24
  - exponencial, 22
  - lineal, 21
  - no determinista, 24
  - polinomial, 21
- complejidad (efectividad), 20
- en tiempo constante, 21
- arista, 2, 41
  - directa, 42
  - esencial, 17
  - inversa, 42
- adyacentes, 2
- capacidad de -, 41
- coste de -, 41
- incidentes, 2
- partición de una -, 10
- asa, 8
- asignación
  - de verdad, 28
    - - satisfactoria, 29
  - ortogonal, 35, 46
    - - óptima, 61
    - - realizable, 80
    - - dirigida, 81
- segmento de - -, 47
- bloque, 5
- bolsillo, 101
- booleana (variable), 28
  - literal, 28
- camino, 3
  - simple, 4
- cara, 9
  - frontera de una -, 9
- ciclo, 4
  - aumentador de flujo, 42
    - coste de - -, 44
  - esencial, 17, 40
  - simple, 4
- cilindro, 37
  - círculos máximos, 38
  - eje, 37
  - generatrices, 38
  - malla cilíndrica, 38
- cláusula, 29
  - satisfacible, 29
- complejidad, 20
  - constante, 21

- exponencial, 22
- lineal, 21
- peor caso, 21
- polinomial, 21
- completo, 2
  - grafo bipartito -, 3
  - grafo -  $(K_n)$ , 2
- conexo, 5
  - componente conexa, 5
- corte
  - conjunto de -, 6
  - pareja de -, 5
  - punto de -, 5
- dibujo, 9
  - ortogonal, vi
  - rectilíneo, vii
- digrafo, 2, 82
  - fuelle, 7
  - st-grafo esférico, 19
  - sumidero, 7
- dual
  - grafo -, 9
- equivalente (inmersión), 12, 17
  - esencialmente -, 141
- Euler, 11
  - característica de -, 11
  - fórmula de -, 11
- flujo, 41
  - ortogonal, 52
  - - óptimo, 61
  - - realizable, 80, 103
- ciclo aumentador de -, 42
  - coste de - -, 44
  - coste de -, 42
  - valor de -, 42
- fuelle, 7, 19, 41
- género, 8
- grafo, 2
  - $k$ -ortogonal, 73
  - 2-conexo, 5
  - 3-conexo, 5
  - 4-plano, 9
  - bar-representable, 18
  - bipartito, 3
  - cíclico, 4
  - cilíndrico, 40
  - completo, 2
  - de bar-visibilidad, 18
  - esférico, 9
    - st-grafo esférico, 19
  - homeomorfo a, 10
  - inmerso, 9
  - isomorfo a, 6
  - $k$ -conexo, 6
  - plano, 9
  - regular, 2
  - $k$ -grafo, 2
  - $k$ -valente, 3
  - subdivisión de un -, 10
- homeomorfismo, 10
- homomorfismo, 6
- inmersión, 9

- 2-celular, 8
- cilíndrica, 73
  - - esférica ascendente, 87
- combinatoria, 11, 16
- equivalente a, 12
- ortogonal, 17, 139, 157
  - - cilíndrica, 35, 40, 73
  - codo, 17
- rectilínea, 142, 159
- topológica, 9, 16
- cara de una -, 9
- ciclo esencial en una -, 17
- rectilínea, 17
- sistema de rotación, 12
- isomorfismo, 6
  - grafos isomorfos, 6
- Kuratowski, 10
  - Teorema de -, 10
- mallá, 16
  - cilíndrica, 38
  - inmersión en la -, 17
    - codo, 17
- meridiano, 13, 15, 17
- multigrafo, 2
  - arista múltiple, 2
  - lazo, 2
- not-all-equal-3sat, 30, 159
- NP (clase de problemas), 23, 25
- NP-completo, 22, 23, 26, 144, 159
- NP-duro, 28, 153, 164
- ortogonal
  - asignación -, 35, 46
    - - óptima, 61
    - segmento de - -, 47
  - dibujo -, vi
  - flujo -, 52
    - - óptimo, 61
  - grafo  $k$ -ortogonal, 73
  - inmersión -, 17, 139, 157
    - - cilíndrica, 35, 73
    - codo, 17
  - representación - estándar, ix, 15, 16
  - mallá, 16
  - rectángulo, 13, 15, 17
  - ventana, 13, 15
- P (clase de problemas), 24
- paralelo, 12, 15, 17
- plano (grafo), 9
  - 4-plano, 9
- polinomial (tiempo)
  - algoritmo -, 21
  - reducción -, 23, 26
  - reducción de Turing -, 28
- Polo Norte, 38
- Polo Sur, 38
- problema
  - NP-completo, 22, 26, 144, 159
  - NP-duro, 28, 153, 164
  - P, 25
  - de 3-satisfacibilidad, 29, 144
  - de búsqueda, 28

- de decisión, 23
- de satisfacibilidad, 23, 28
- intratable, 23
- not-all-equal-3sat, 30, 159
- punto, 41
- realizable
  - asignación ortogonal -, 80
    - - bidireccional, 97
    - - unidireccional, 97
  - ciclo aumentador de coste negativo -, 103
  - flujo ortogonal -, 80, 103
- rectilínea
  - inmersión -, 17, 142, 159
- red, 41, 49
- reducción, 23
  - de Turing polinomial, 28
  - polinomial, 26
- rotación en un vértice, 12
- satisfacibilidad, 23, 28, 29
- st-numeración, 7
  - fuerza, 7
  - sumidero, 7
- subgrafo, 5
  - inducido, 5
- sumidero, 7, 19, 41
- superficie, 8, 139, 157
  - orientable, 8
  - ortogonal estándar, ix, 15, 16
  - mallado, 16
  - rectángulo, 13, 15, 17
    - ventana, 13, 15
  - poligonal estándar, 8, 12
  - tórica, 8, 12
  - asa, 8
  - género de una -, 8
  - inmersión 2-celular, 8
- toro, 8, 12
  - doble toro, 8
  - toro plano, 13
- vértice, 2
  - adyacentes, 2
  - incidentes, 2
  - rotación en un -, 12
- valencia, 2
- ventana, 13, 15
  - rectángulo, 13, 15, 17
- visibilidad
  - grafo bar-representable, 18
  - grafo de bar -, 18
  - representación de -, 18

