



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

VARIANTES DEL PROBLEMA DE LA GALERÍA DE ARTE Y GEOMETRÍA DE ORIENTACIONES RESTRINGIDAS

TESIS
QUE PARA OPTAR POR EL GRADO DE
DOCTOR EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

PRESENTA:
JESÚS NESTALY MARÍN NEVÁREZ

DIRECCIÓN:
JORGE URRUTIA GALICIA
INSTITUTO DE MATEMÁTICAS, UNAM

COMITÉ TUTOR:
CARLOS BRUNO VELARDE VELÁZQUEZ
INSTITUTO DE INVESTIGACIONES EN MATEMÁTICAS APLICADAS Y EN SISTEMAS, UNAM

JOSÉ DAVID FLORES PEÑALOZA
FACULTAD DE CIENCIAS, UNAM

CD. DE MÉXICO, JUNIO DE 2022



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

Al Consejo Nacional de Ciencia y Tecnología (CONACYT), a la Universidad Nacional Autónoma de México, al Posgrado en Ciencia e Ingeniería de la Computación, al Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas de la UNAM, al Instituto de Matemáticas de la UNAM, al Programa de Apoyo a los Estudios de Posgrado (PAEP) y al Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica (PAPIIT).

A mis sinodales, el Dr. Jorge Urrutia Galicia, el Dr. Carlos Bruno Velarde Velázquez, el Dr. José David Flores Peñaloza, el Dr. Ruy Fabila Monroy y el Dr. Juan José Montellano Ballesteros, por tomarse el tiempo de revisar este trabajo y por sus valiosos comentarios.

A mi supervisor, el Dr. Jorge Urrutia, por toda la motivación y el apoyo que me ha brindado durante mi doctorado, y por su disposición para continuar trabajando durante la pandemia, ya que logramos ser productivos a pesar de estar pasando por tiempos difíciles.

Al Dr. Carlos Velarde, por su exhaustiva revisión de este trabajo, la cual derivó en valiosas recomendaciones y correcciones que mejoraron considerablemente su calidad.

A la Dra. Adriana Ramírez Viguera, a José Luis Álvarez Rebollar y a Oriol Solé Pi por todas estas tardes de trabajo, las cuales han hecho más llevadera la pandemia.

A mis padres Jesús y Yolanda, y mis hermanos Diana y Alejandro, son para mí una fuente de inspiración y los llevo siempre en mi corazón.

A Azalia, mi novia, por todo su apoyo durante estos cuatro años, es la mejor compañía que podría haber tenido durante esta pandemia.

Y a cualquier otra persona que me apoyara y que haya omitido en las líneas anteriores. ¡Muchas gracias!

Índice general

1. Introducción	5
1.1. Inicios de la Geometría Computacional	6
1.1.1. Enfoque moderno: década de 1970	8
1.1.2. De los resultados teóricos a los algorítmicos: dibujar gráficas planas	10
1.1.3. Principales aportes del trabajo doctoral de Shamos	11
1.2. El Problema de la Galería de Arte	12
1.3. Organización y contribuciones de esta tesis	13
2. Protección de islas con drones recargables	15
2.1. Introducción	15
2.2. Definiciones	16
2.3. Trabajo relacionado	19
2.4. Resultados preliminares	21
2.5. Algoritmo	23
2.5.1. Análisis de complejidad	25
3. Cobertura de ortoárboles con balizas	35
3.1. Introducción	35
3.2. Trabajo relacionado	36
3.3. Definiciones	37
3.4. Resultados auxiliares	38
3.5. Cobertura de ortoárboles con balizas	43
3.6. Complejidad del problema de la galería de arte en ortoárboles	46
4. Incrustaciones de árboles en mallas rectangulares	49
4.1. Introducción	49
4.2. Definiciones	51

4.3. Trabajo relacionado	52
4.4. Resultados	53
4.4.1. Algoritmo	58
4.5. Área del encaje	62
5. Cuadriláteros arcoiris en conjuntos de puntos coloreados	63
5.1. Introducción	63
5.1.1. Definiciones	64
5.2. Trabajo previo	65
5.3. Cuadriláteros arcoiris de área mínima o máxima	67
5.3.1. Algoritmo	70
5.4. Cuadriláteros arcoiris vacíos	73
6. Conclusiones	76
Bibliografía	80

Capítulo 1

Introducción

Esta tesis es un compendio de resultados para algunos de los problemas en los que trabajé durante mi doctorado bajo la dirección del Dr. Jorge Urrutia. Estos problemas pertenecen al área de geometría computacional, por lo que comenzaremos esta tesis con una breve introducción a esta disciplina.

Puesto en términos simples, en el área de la geometría computacional tenemos como objetivo diseñar y analizar algoritmos para resolver problemas de tipo geométrico de manera eficiente. Además, queremos determinar aquellos problemas que no tienen una solución en tiempo polinomial o que no son computables. En caso de que un problema no tenga una solución polinomial podríamos estar interesados en dar información combinatoria sobre su solución, o bien, en obtener una solución aproximada a la óptima. Al centro de este campo de estudio tenemos un conjunto de técnicas algorítmicas que se han ido desarrollando a lo largo del tiempo. Estos algoritmos utilizan una colección de estructuras de datos, algunas de las cuales ya son conocidas en computación, y otras que han sido desarrolladas específicamente con el fin de resolver problemas geométricos.

La geometría computacional establece conexiones entre la geometría discreta, la teoría de la computación y aplicaciones prácticas. Muchos de los problemas de la vida real que se quieren modelar con estas herramientas tienen aspectos geométricos, los cuales necesitamos considerar para poder encontrar soluciones eficientes. Al mismo tiempo, las aplicaciones han servido de inspiración para plantear nuevos problemas geométricos, como es el caso del problema del Capítulo 2, en el que tomamos en cuenta las limitaciones físicas de vehículos aéreos no tripulados.

1.1. Inicios de la Geometría Computacional

Es común encontrar en libros o recopilaciones relacionadas con el área la afirmación de que el estudio de la geometría computacional, tal como la conocemos hoy en día, se inició a principios la década de 1970. Sin embargo, desde el siglo anterior se habían obtenido resultados que ahora son considerados clásicos en esta área y en la de geometría discreta. Estos resultados siguen teniendo relevancia, pues actualmente se estudian nuevos problemas relacionados o que están inspirados en ellos. A continuación mencionamos algunos ejemplos relevantes:

El problema de encontrar el círculo más pequeño que encierra a un conjunto de n puntos en el plano fue planteado por James J. Sylvester en 1857 [91]. La primera solución conocida a este problema fue presentada por George Chrystal en 1885 [26], y su solución óptima, un algoritmo de tiempo lineal basado en programación lineal, fue presentada por Nimrod Megiddo en 1983 [75]. El diagrama de Voronoi de dimensión n fue definido por Georgy Fedosievych Voronoy en 1908 [104], y el primer algoritmo óptimo, de tiempo $O(n \log n)$ para obtenerlo en el plano fue presentado en 1975 por Michael Shamos y Dan Hoey [88]. El Teorema de Helly sobre intersección de convexos fue presentado por Eduard Helly en 1930 [63]. La triangulación de Delaunay fue introducida por Borís Delaunay en 1934 [38]. En 1935 Paul Erdős y George Szekeres [49] respondieron afirmativamente la siguiente pregunta: ¿Es posible encontrar para un n dado, un número $N(n)$ tal que, para cualquier conjunto que contenga al menos $N(n)$ puntos, sea posible seleccionar n de ellos que formen un polígono convexo? Encontrar el valor exacto de $N(n)$ es un problema que ha probado ser sumamente difícil. Se han estudiado muchas variantes de este problema desde entonces, incluyendo variantes en conjuntos de puntos coloreados como las estudiadas por Devillers et al. en 2003, véase [39]. En 1948 Itsván Fary [51] demostró que cualquier gráfica aplanable puede ser dibujada sin cruces en sus aristas y representando cada arista como un segmento de recta. Dicho de otra forma, usar curvas en lugar de segmentos para representar las aristas no permite dibujar una clase más grande de gráficas. Por último, tenemos dos algoritmos famosos en gráficas con pesos asignados a las aristas. En 1957 Robert Prim presentó un algoritmo para encontrar un árbol generador de peso mínimo [82], mientras que en 1959 Edsger Dijkstra presentó un algoritmo para encontrar la trayectoria más corta desde un vértice de origen al resto de los vértices [42]. Estos algoritmos siguen siendo estudiados hasta el día de hoy en los cursos de algoritmos.

La geometría computacional podría considerarse una disciplina anterior incluso a los siglos diecinueve y veinte. Algunos autores sostienen que los griegos obtuvieron

soluciones para problemas que podríamos considerar como pertenecientes a esta área. Godfried Toussaint [99] menciona el siguiente ejemplo:

Herón de Alejandría, quien vivió aproximadamente entre los años 10 y 75 de la era común, estudió el siguiente problema. Sea ℓ una línea en el plano y sean p y q dos puntos del mismo lado con respecto a ℓ . Queremos encontrar un punto r en ℓ tal que, si queremos ir de p a q siguiendo una trayectoria que pase por r , se minimice la longitud de dicha trayectoria. Cuatrocientos años antes, Euclides había postulado dos principios sobre la reflexión de la luz: el plano de incidencia de la luz coincide con su plano de reflexión, y el ángulo de incidencia de la luz coincide con su ángulo de reflexión. Basándose en estos principios, Herón realizó el siguiente postulado en su obra *Catóptrico*: la luz siempre viaja a lo largo de la trayectoria más corta. La solución a este problema, conocida como el Teorema de Herón, dice lo siguiente. Si queremos ir de p a q colocados del mismo lado de ℓ , por medio de la trayectoria más corta que toque un punto r en ℓ , entonces tomamos un punto q' que sea el reflejo de q del lado opuesto de ℓ , y después definimos a r como la intersección de ℓ con el segmento que une a p y q' . Se muestra una ilustración de esto en la Figura 1.1. Utilizando la desigualdad del triángulo se prueba que para cualquier otro punto r' en ℓ obtenemos una trayectoria más larga. Estos principios siguen teniendo aplicaciones en nuestros días. Un par de ejemplos de estas aplicaciones son las soluciones para dos problemas relacionados con el que estudiamos en el Capítulo 2. El primero es la solución al Problema de la Ruta del Vigilante (del inglés, *Watchman Route Problem*) en polígonos ortogonales dada por Chin y Ntafos en 1986 [24]. El segundo es la solución al Problema del Vigilante del Acuario (del inglés, *Aquarium Keeper's Problem*) obtenida por Czyzowicz et al. en 1991 [33].

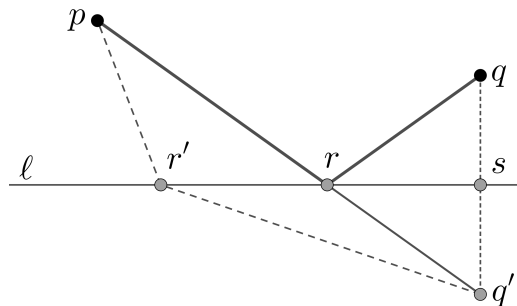


Figura 1.1: Ilustración de la demostración del Teorema de Herón. La distancia de q a s es igual que la de q' a s , mientras que la distancia de p a q' es menor que la de p a r' más la de r' a q' .

1.1.1. Enfoque moderno: década de 1970

Es común encontrar en textos dedicados a la geometría computacional, como en libros o recopilaciones, la afirmación de que esta disciplina, tal y como la conocemos en el presente, fue desarrollada en la década de 1970. Algunos autores atribuyen el inicio de este nuevo enfoque a publicaciones como la tesis de doctorado de Michael Shamos, titulada simplemente *Computational Geometry* [86]; la tesis de Forrest, titulada *Curves and Surfaces for Aided Design* [54], y el artículo de Ron Graham de 1972 en el que se presenta un algoritmo de tiempo $O(n \log n)$ para encontrar el cierre convexo de un conjunto de n puntos [59]. Cabe resaltar que un algoritmo similar al de Graham fue publicado en 1967 por Bass y Schubert [10], pero en este artículo no se presentó un análisis de complejidad. Toussaint [99] menciona que, si bien Shamos no inició (al menos por sí solo) el enfoque moderno de la geometría computacional, su contribución principal a este campo fue el énfasis en incluir, con cada algoritmo, un análisis de complejidad en términos de la notación O grande (del inglés, *Big O*), además de la introducción de las cotas inferiores en el análisis de complejidad de problemas geométricos. Además, en el trabajo inicial de Shamos comienza a hacerse más notoria la diferencia entre conocer propiedades teóricas sobre objetos geométricos, y poder utilizar esas propiedades para resolver problemas de manera eficiente. Con la intención de poner en contexto la manera en la que se dio este cambio de enfoque, hablaremos sobre algunos de los primeros problemas en los que Michael Shamos trabajó durante sus estudios doctorales en la década de 1970, según los relata en sus memorias respecto al tema [87].

El primero es un problema relacionado al que estudiamos en el Capítulo 4, y para el cual no se había obtenido aún una solución algorítmica: dibujar estructuras químicas en la pantalla de forma presentable para la vista humana. Estas estructuras se pueden representar mediante gráficas, lo cual hace a este problema equivalente al de encontrar dibujos (encajes) de estas gráficas que brinden suficiente información al ojo humano. Por lo anterior, se hizo necesario observar a las gráficas no solamente como objetos combinatorios, sino también como objetos geométricos. Se dice que una gráfica es *aplanable* si puede ser dibujada en el plano de modo que sus vértices sean representados por puntos y sus aristas por curvas con interiores disjuntos dos a dos (es decir, sin cruces). Fary [51] había probado décadas antes que toda gráfica aplanable se puede dibujar en el plano sin cruces de modo que sus aristas sean segmentos de línea recta. Lamentablemente, este resultado por sí solo no brinda una estrategia clara para conseguir dicho encaje. De hecho, dada una gráfica se requiere un paso anterior a buscar su encaje: determinar si es aplanable. Si para este fin aplicamos directamente la caracterización de Kuratowski [69], que dice que una

gráfica es aplanable si y sólo si no contiene una subgráfica homeomorfa a K_5 o $K_{3,3}$, obtendríamos un algoritmo de complejidad temporal al menos $O(n^5)$. Sin embargo, mientras trabajaba en este problema, Shamos se encontró con un artículo de Robert Tarjan [97] en el que se presenta un algoritmo de tiempo lineal para determinar si una gráfica dada es aplanable. Este algoritmo consigue la complejidad lineal gracias a que no se basa directamente en ninguno de los teoremas clásicos para caracterizar las gráficas aplanables. Aunque el problema de encajar gráficas aplanables quedó pendiente por varios años más, este primer encuentro dejó una importante lección: para desarrollar algoritmos eficientes en problemas geométricos hay que ir más allá de las técnicas que se utilizaban en las matemáticas tradicionales.

Tal vez a falta de este enfoque es que, en ese tiempo, muchos problemas que hoy consideramos problemas básicos seguían sin ser resueltos. Una colección publicada por Donald Knuth en 1973 [67] contenía problemas como el siguiente. Dados n puntos en el plano, ¿qué tan rápido se puede determinar cuál de ellos es el más cercano a un nuevo punto p ? Shamos trabajó en un par de problemas relacionados: dados n puntos en el plano, encontrar los dos puntos más lejanos y encontrar los dos puntos más cercanos. El segundo problema llevaría eventualmente al desarrollo de un algoritmo $O(n \log n)$ para computar el diagrama de Voronoi. En 1973 Shamos diseñó un algoritmo de tiempo lineal para encontrar el diámetro de un polígono convexo. Al utilizar el algoritmo de Ron Graham para obtener, como paso previo, el cierre convexo de un conjunto de puntos, el algoritmo de Shamos permite encontrar la pareja de puntos más lejana en un conjunto de n puntos en tiempo $O(n \log n)$. Según Shamos [87], el primer análisis de complejidad en un problema de geometría computacional se le atribuye a Graham justo por este artículo sobre encontrar el cierre convexo de un conjunto de puntos [59]. Sin embargo, aunque Graham utilizó la notación O grande para describir la cota superior, no menciona la cota inferior de complejidad temporal para este problema, la cual se establecería en 1985 [81]. Por otro lado, seguiría sin haber resultados por varios años en el problema de encontrar el par de puntos más cercanos en un conjunto de n puntos.

Un aspecto que hoy consideramos básico en geometría computacional es el siguiente: Al tomar en cuenta una estructura geométrica en problemas que previamente se habían estudiado de forma combinatoria pueden surgir diferencias considerables. Esto recién se comenzaba a notar entre quienes se incursionaban en este campo durante la primera mitad de la década de 1970. Shamos y su compañero Dan Hoey intentaron resolver el problema de encontrar un árbol generador euclidiano mínimo dada una gráfica geométrica. Después de hacer experimentos con computadora, una técnica cuyo uso no era todavía común para resolver este tipo de problemas, lograron

encontrar varias diferencias con respecto al caso general. Es sabido que en gráficas combinatorias pueden existir $n(n-1)/2$ aristas, lo que lleva a un algoritmo de complejidad temporal $O(n^2 \log n)$. Con sus experimentos, Shamos y Hoey encontraron que el grado máximo de cualquier vértice en el caso geométrico es seis. Este hecho es sencillo de demostrar cuando ya se conoce, pero su prueba no había sido publicada aún en alguna revista (aunque ya había aparecido en un memorándum interno de *Bell Laboratories* escrito por Graham). En ese momento no lograron encontrar un algoritmo para resolver el problema; después se demostraría que el árbol generador euclidiano mínimo es subgráfica de la triangulación de Delaunay y que puede ser obtenido en tiempo $O(n \log n)$.

1.1.2. De los resultados teóricos a los algorítmicos: dibujar gráficas planas

Regresando al problema de encontrar encajes de gráficas aplanables, la solución de Fary [51] requería encontrar repetidamente un punto en el interior del núcleo (del inglés, *kernel*) de un polígono *estrellado* (un polígono que contiene una región no vacía llamada núcleo tal que cada uno de sus puntos puede ver la totalidad del polígono). En un artículo de Yaglom y Boltyanskii [17] se dedujo que el núcleo de un polígono puede ser obtenido al intersectar un conjunto de semiplanos definido por sus aristas. Sin embargo, en 1973 no había información en la literatura disponible en la Universidad de Yale, donde Shamos estudiaba, acerca de algún algoritmo para encontrar la intersección de un conjunto de semiplanos; ni siquiera parecía existir en la literatura un buen algoritmo para detectar si dos segmentos de línea se intersectan. Al parecer, para poder resolver este tipo de problemas se tenían que desarrollar nuevas herramientas, lo cual desde el punto de vista del tesista puede parecer un gran obstáculo, pero que visto de una manera más optimista, aportaba al argumento de que la geometría computacional era un área de estudio aparte.

Puede ser que, por esa misma falta de herramientas, la primera solución obtenida para este problema, descrita a continuación, es vista ahora (incluso por el mismo Shamos [87]) como una solución de fuerza bruta, a pesar de que en su tiempo fuera una novedad enfrentar así este tipo de problemas. Usando la observación de que la intersección de un conjunto de semiplanos es convexa procedemos como sigue: Primero obtenemos la intersección de los primeros dos semiplanos. Después, intersecamos repetidamente la solución actual con el siguiente semiplano. Esto resulta en un algoritmo de tiempo $O(n^2)$ para obtener el núcleo de un polígono estrellado y, al aplicar este resultado a la solución de Fary, se obtiene un algoritmo de tiempo $O(n^3)$ para

encontrar un encaje rectilíneo de una gráfica aplanable. En palabras de Shamos [87] este algoritmo era sumamente crudo, pero no conocía herramienta alguna que ayudara a mejorarlo. Además, en ese tiempo no se conocía un problema computacional con a lo más n entradas y n salidas cuya cota inferior de complejidad temporal fuera peor que $\Omega(n \log n)$, pero cuya solución fuera de tiempo polinomial, por lo que quedaba abierto el problema de cerrar estas cotas.

1.1.3. Principales aportes del trabajo doctoral de Shamos

Con la idea de formular problemas que parecieran interesantes, y cuya solución pudiera ser utilizada para resolver otros problemas del área, Shamos comenzó a pensar en aquellos que no hubiesen sido aún resueltos desde un punto de vista computacional. Por ejemplo, dado un polígono P , ¿cuál es el triángulo más grande contenido en P ?, ¿cuál es el triángulo más pequeño que contiene a P ?, ¿es P simple?, ¿cuál es el menor número de polígonos convexos disjuntos cuya unión es P ?, ¿es P congruente a otro polígono dado Q ? Para 1974, Shamos tenía una lista de 75 problemas, los cuales describió en un manuscrito que tituló *Problems in Computational Geometry* [85], en el que incluyó el estado del arte sobre los problemas y los algoritmos obvios para resolverlos. Un objetivo de esta colección era promover la idea antes mencionada de que la geometría computacional es un área de estudio aparte y no solo una colección de unos cuantos problemas que parecen tener relación entre sí.

En el mismo año 1974, Hoey le presentó a Shamos una descripción de la subdivisión del plano generada al tomar una colección de n puntos, y después considerar las regiones que contienen los puntos del plano más cercanos al mismo elemento de dicha colección. Al revisar la literatura encontraron que esta subdivisión del plano había sido descrita en 1908 por Voronoi [104], pero no encontraron una forma de construirla. Eventualmente, consiguieron un algoritmo de tiempo $O(n \log n)$ para este problema, el cual fue presentado en [88]. En dicho artículo se incluyeron varias aplicaciones, una de las cuales era que el problema de los dos puntos más cercanos puede ser resuelto a través del diagrama de Voronoi. Además, probaron que esta complejidad es una cota inferior para este problema, por lo que su algoritmo era óptimo. Este resultado permitiría que Shamos consiguiera su tesis doctoral, aunque esta sería publicada hasta 1978 [86].

Años después de que Shamos obtuviera su grado, Ron Graham intercedería para que en 1985 publicara junto con Franco Preparata un libro de geometría computacional titulado *Computational Geometry: an introduction* [81]. Este libro se convirtió un referente para introducirse en la materia, y cabe resaltar que Shamos atribuye

la mayor parte de su escritura a Preparata. Otro libro de importancia para el área, *Algorithms in Combinatorial Geometry* [44], cuyo autor es Herbert Edelsbrunner, fue publicado un par de años después.

1.2. El Problema de la Galería de Arte

El Problema de la Galería de Arte y sus variantes constituyen un área de la geometría computacional estudiada extensamente en las últimas décadas. Este problema fue planteado en 1973 por Victor Klee de la siguiente manera: dada una galería de arte de forma arbitraria, modelada como un polígono simple con n lados, ¿cuál es el mínimo número de guardias requerido para que esta sea vigilada? [64]. Esta pregunta fue respondida por Václav (Vašek) Chvátal en 1975 [27], quién probó las cotas combinatorias para este problema en una respuesta conocida como el Teorema de la Galería de Arte: $n/3$ guardias son siempre suficientes y a veces necesarios para vigilar una galería de arte con n paredes. En 1981, Avis y Toussaint presentaron un algoritmo de tiempo $O(n \log n)$ para encontrar la ubicación de los guardias, la cual también permite descomponer el polígono en piezas con forma de estrella.

El Teorema de la Galería de Arte es una respuesta combinatoria, pero que puede dar una respuesta muy ineficiente en polígonos en los que el tamaño de la solución óptima es muy pequeño con respecto a la cantidad de paredes de la galería. Sin embargo, el problema de encontrar el conjunto mínimo de guardias que vigile un polígono no tiene una solución exacta en tiempo polinomial. Este hecho se mantiene aún en clases restringidas de polígonos. Por ejemplo, Ana Paula Tomás [98] demostró que el problema de la galería de arte es NP-difícil en una familia de polígonos ortogonales llamados *polígonos delgados*. Estos polígonos que tienen la particularidad de que la gráfica dual de la subdivisión obtenida al extender sus aristas hacia el interior es un árbol.

En 1978, Steve Fisk [52] presentó una demostración sumamente elegante para el Teorema de la Galería de Arte en un artículo de una sola página. Esta demostración consiste en triangular el polígono y colorear con tres colores los vértices de esta triangulación de manera que cualesquiera dos vértices adyacentes tengan colores distintos. Así, dado que cada triángulo incide en un vértice de cada color, podemos vigilar el polígono con la clase cromática de menor cardinalidad, que tendrá a lo más $n/3$ vértices. Para una ilustración de esta demostración ver la Figura 1.2. La demostración que presentamos para el problema de vigilancia de ortoárboles con balizas en el Capítulo 3 se basa en la idea de Fisk de seleccionar una subdivisión del polígono, asignar un conjunto de vértices a cada pieza y colorear estos vértices.

Esta demostración difiere de la de Fisk en que la subdivisión que obtenemos no es una partición sino una cobertura, es decir, los interiores de piezas adyacentes no son necesariamente disjuntos en nuestra prueba.

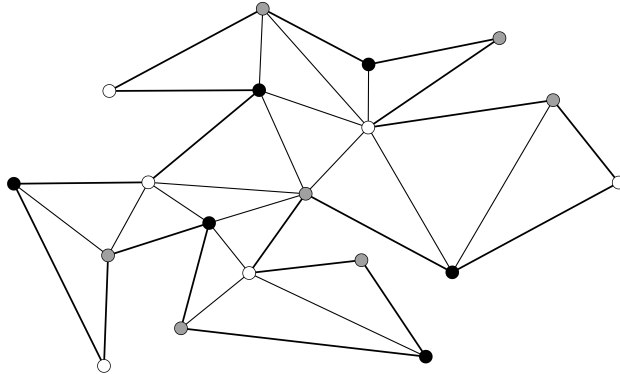


Figura 1.2: Ilustración de la demostración Steve Fisk para el Teorema de la Galería de Arte.

Para conocer más sobre las numerosas variantes del Problema de la Galería de Arte y problemas relacionados con visibilidad el lector es referido a los libros de Joseph O'Rourke [80] y Subir Ghosh [57], así como las recopilaciones de Thomas Shermer [89] y Jorge Urrutia [100].

1.3. Organización y contribuciones de esta tesis

Los primeros dos capítulos de esta tesis están dedicados a variantes del Problema de la Galería de Arte. En el Capítulo 2 estudiamos un problema de vigilancia en el que queremos colocar un conjunto de estaciones de carga en la frontera de una isla, de manera que un dron pueda darle la vuelta completa al usar dichas estaciones para recargar su batería. Este resultado forma parte de un artículo enviado a la revista *Theoretical Computer Science* de la editorial Elsevier bajo el título *Optimal placement of base stations in border surveillance using limited capacity drones*. A la fecha, este artículo ha sido aceptado.

En el Capítulo 3 estudiamos una variante del Problema de la Galería de Arte en \mathbb{R}^3 en el que queremos vigilar una clase de poliedros ortogonales llamados ortoárboles con un tipo de guardias llamados *balizas*. Una baliza vigila a todos los puntos desde los que se puede atraer un objeto de manera similar a como un imán atrae partículas ferrosas. El resultado que se presenta en este capítulo es continuación del trabajo

de tesis de maestría del autor [74], en la que se consiguió la cota inferior para el problema, pero no la cota superior. Este trabajo se publicó en la revista *Graphs and Combinatorics* de la editorial Springer bajo el título *Tight bounds for illuminating and covering of orthotrees with vertex lights and vertex beacons* [4].

En el Capítulo 4 estudiamos el siguiente problema sobre encontrar encajes de gráficas en el plano: Dado un árbol T , queremos encontrar un encaje en el que los vértices sean puntos con coordenadas enteras y las aristas segmentos de recta verticales u horizontales. Además, queremos que se minimice el número máximo de dobleces de cualquier trayectoria de T en este encaje, es decir, el número de veces que una arista vertical es adyacente a una horizontal en una trayectoria de T . Este encaje puede ser obtenido en tiempo lineal y el algoritmo se puede adaptar para obtener un encaje de un árbol de grado d en dimensión $\lceil d/2 \rceil$ manteniendo la complejidad temporal lineal. Este trabajo fue publicado en la revista *Information Processing Letters* de la editorial Elsevier bajo el título *Grid straight-line embeddings of trees with a minimum number of bends per path* [73].

Por último, en el Capítulo 5 estudiamos dos problemas en conjuntos de puntos coloreados en el plano. Estos problemas consisten en encontrar polígonos arcoíris en estos conjuntos de puntos, es decir, polígonos tales que cada uno de sus vértices sea un elemento del conjunto con color distinto al resto de los que componen al polígono. En el primer problema queremos encontrar un cuadrilátero arcoíris de menor área o de mayor área en un conjunto de puntos coloreado con k colores, para el cual presentamos un algoritmo de tiempo $O(n^2k)$. El segundo problema trata sobre encontrar un cuadrilátero arcoíris vacío, ya sea convexo o no, y presentamos un algoritmo de complejidad temporal $O(n^3)$ para resolver este problema. Este algoritmo busca primero la existencia de un cuadrilátero arcoíris vacío convexo, y en caso de que no exista, reporta si existe uno que no sea convexo. Un artículo que incluye estos resultados además de otros que no fueron incluidos en esta tesis fueron enviados a la revista *Graphs and Combinatorics* de la editorial Springer bajo el título *On rainbow quadrilaterals in colored point sets*. Los resultados no incluidos pueden ser encontrados en la tesis de maestría de Alma Arévalo [7], mientras que implementaciones de algunos de estos algoritmos se presentaron en la tesis de maestría de Rodrigo Chávez [21].

Durante mi doctorado trabajé con mi director de tesis y mis compañeros en problemas que no fueron incluidos en esta tesis y que derivaron en publicaciones de revista. Estas publicaciones son: *Finding minimum witness sets in orthogonal polygons*, publicado en *Computational Geometry: theory and applications* [2], y el artículo *Representing point sets on the plane as permutations* publicado en *Information Processing Letters* [6], ambas revistas de la editorial Elsevier.

Capítulo 2

Protección de islas con drones recargables

2.1. Introducción

En esta sección estudiamos una variante del problema de la galería de arte que llamamos *problema del mínimo número de estaciones*: Supongamos que queremos proteger la frontera de una isla \mathcal{I} cuya frontera está modelada como un polígono simple \mathcal{P} por medio de un conjunto de vehículos aéreos no tripulados, a los que llamaremos *drones*. Consideramos que los drones tienen un *rango de vuelo* d relativamente corto entre recargas. Además, tendrán permitido volar solamente sobre la frontera de \mathcal{I} o sobre el mar que la rodea. El objetivo es colocar un conjunto $S = \{s_0, \dots, s_{k-1}\}$ de estaciones de recarga de cardinalidad mínima k sobre la frontera de \mathcal{I} tal que, al visitar todos los elementos de S en orden y regresar a la estación inicial s_0 , cada dron viaje alrededor de \mathcal{P} sobre una curva cerrada \mathcal{C} . Para satisfacer la restricción del rango de vuelo de los drones, la solución obtenida debe cumplir que el intervalo de \mathcal{C} entre cada par de estaciones consecutivas s_i y s_{i+1} (adición módulo k) tenga longitud a lo más d . En la Figura 2.1 se muestra un ejemplo de una isla con un conjunto de cardinalidad mínima de estaciones de recarga; notemos que en una solución de cardinalidad mínima las estaciones no necesariamente están colocadas sobre el cierre convexo de \mathcal{P} . Decimos que el conjunto $S = \{s_0, \dots, s_{k-1}\}$ (de cardinalidad mínima) es una *solución óptima* para este problema y que un conjunto $S' = \{s'_0, \dots, s'_k\}$ con $k + 1$ estaciones de recarga es una solución *casi óptima* para este problema.

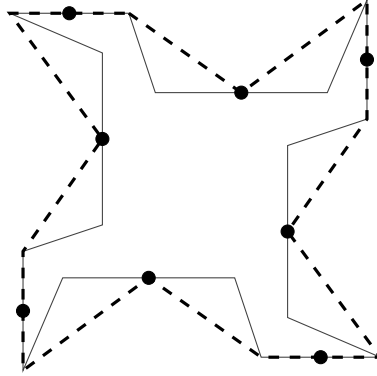


Figura 2.1: Un polígono \mathcal{P} , un conjunto mínimo de estaciones representados como puntos, y la curva cerrada que siguen los drones representada como una línea discontinua.

Sea \mathcal{P} un polígono con n vértices y sea L el perímetro de \mathcal{P} . Presentamos un algoritmo de complejidad temporal $O(n^2 + \frac{L}{d}n)$ para la siguiente versión restringida del problema del mínimo número de estaciones. Dado un punto s_0 en el cierre convexo de \mathcal{P} , $CH(\mathcal{P})$, el algoritmo regresa una solución óptima bajo la restricción de que s_0 pertenezca al conjunto S . La solución producida por nuestro algoritmo es una solución óptima o casi óptima para la versión sin restricciones del problema del mínimo número de estaciones. Notemos que el problema de encontrar una solución óptima para el problema del mínimo número de estaciones sin restricciones es entonces equivalente a encontrar la posición de una sola estación de la solución óptima, lo cual quedará como un problema abierto.

2.2. Definiciones

Definimos un *polígono* \mathcal{P} como una sucesión de n vértices p_0, p_1, \dots, p_{n-1} cuyas aristas son los segmentos $s(p_i, p_{i+1})$, adición módulo n . Todos los polígonos con los que trabajamos son *simples*, es decir, los pares de aristas no consecutivas tienen intersección vacía. La *frontera* de un polígono \mathcal{P} , denotada por $\partial\mathcal{P}$, es la curva cerrada obtenida de la unión de todas sus aristas. Notemos que $\partial\mathcal{P}$ divide el plano en dos regiones: el interior de \mathcal{P} , denotado por $Int(\mathcal{P})$, y el exterior de \mathcal{P} , denotado por $Ext(\mathcal{P})$. Decimos que \mathcal{P} está orientado en *dirección horaria* si existe un $\epsilon > 0$ tal que cualquier punto contenido a la derecha de la arista dirigida de p_i a p_{i+1} , adición tomada módulo n , y a distancia menor o igual que ϵ de dicha arista, pertenece a $Int(\mathcal{P})$. El *perímetro* o *longitud* de \mathcal{P} es la suma de las longitudes de sus aristas. Una

trayectoria poligonal es una secuencia de vértices q_0, q_1, \dots, q_m junto con el conjunto de aristas, $s(q_i, q_{i+1})$, $i = 0, \dots, m < n$, y su longitud es la suma de las longitudes de sus aristas.

Dados dos puntos distintos $a, b \in \partial\mathcal{P}$, el *intervalo* $[a, b]$ es el conjunto de puntos de $\partial\mathcal{P}$ visitados al movernos sobre $\partial\mathcal{P}$ en dirección horaria desde a hasta b , ver la Figura 2.2 para un ejemplo. Notemos que, como a y b son distintos, $[a, b] \neq [b, a]$, $[a, b] \cup [b, a] = \partial\mathcal{P}$, y el perímetro de \mathcal{P} es igual a la suma de las longitudes de $[a, b]$ y $[b, a]$.

Denominamos *punte* a un segmento $s(a, b)$ tal que $a, b \in \partial\mathcal{P}$ y su interior está contenido en $Ext(\mathcal{P})$; si la longitud de $s(a, b)$ es a lo más d entonces lo llamaremos un *d-punte* de \mathcal{P} . Notemos que ningún dron puede volar a lo largo de un puente de longitud mayor a d ; además, dado que las estaciones de recarga solo pueden ser colocadas en $\partial\mathcal{P}$, si un dron intenta cruzar un puente de longitud mayor a d , entonces caerá al mar al quedarse sin energía.

Decimos que una trayectoria poligonal que une a dos puntos $a, b \in \partial\mathcal{P}$ es una *d-trayectoria* si todas sus aristas son *d-puentes* de \mathcal{P} o segmentos contenidos en aristas de \mathcal{P} , ver Figura 2.2 para un ejemplo. Un polígono \mathcal{C} es una *d-envoltura* de \mathcal{P} si encierra a \mathcal{P} y todas sus aristas son *d-puentes* de \mathcal{P} o segmentos de aristas de \mathcal{P} . Notemos que existe una cantidad infinita de *d-envolturas* para un polígono \mathcal{P} , entre las cuales está $\partial\mathcal{P}$.

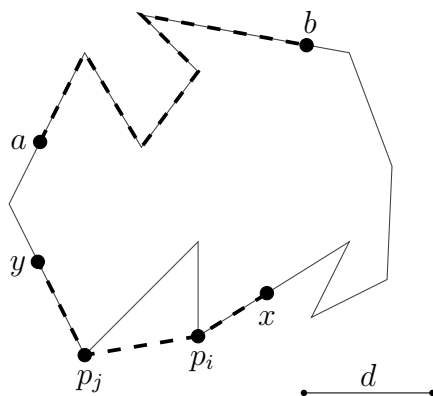


Figura 2.2: El intervalo $[a, b]$ y una *d-trayectoria* $\pi_{x,y}$ que une a los puntos x y y , ambas mostradas con líneas discontinuas. El segmento $s(p_i, p_j)$ es un *d-punte* contenido en $\pi_{x,y}$.

La *distancia de dron* $\delta(a, b)$ del punto $a \in \partial\mathcal{P}$ al punto $b \in \partial\mathcal{P}$ es la longitud de

la d -trayectoria más corta que une a a con b en sentido horario. Por ejemplo, en la Figura 2.2 se muestra la d -trayectoria más corta del punto x al punto y . En adelante nos referiremos a $\delta(a, b)$ como la *distancia* de a a b . Notemos que, en general, $\delta(a, b)$ es distinta de $\delta(b, a)$. Notemos por último que si un dron con rango de vuelo d puede volar del punto a al punto b sin necesidad de recargar su batería, entonces existe una d -trayectoria de longitud a lo más d que los une.

Las definiciones anteriores nos permiten replantear el problema del mínimo número de estaciones de la siguiente manera:

Problema 1 (MinEstaciones). Dados un polígono \mathcal{P} y un valor d , encontrar un conjunto de estaciones de recarga de cardinalidad mínima $S = \{s_0, \dots, s_{k-1}\}$ tal que para cada $0 \leq i \leq k-1$, $s_i \in \partial\mathcal{P}$ y existe una d -trayectoria π_i de longitud a lo más d que une a s_i con s_{i+1} , adición tomada mod k . Además, debe cumplirse que $\mathcal{C} = \pi_0 \cup \dots \cup \pi_{k-1}$ sea una d -envoltura de \mathcal{P} .

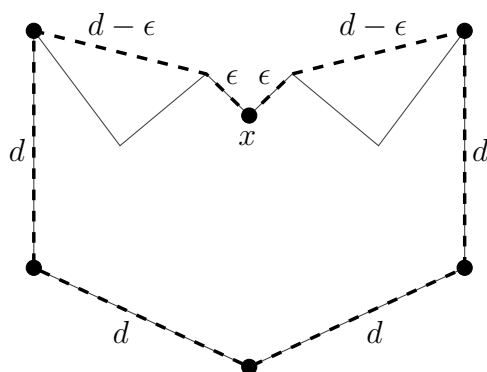


Figura 2.3: Un polígono \mathcal{P} y una solución óptima que consiste en 6 estaciones de recarga. Si se requiere que las estaciones se coloquen sobre la d -envoltura de \mathcal{P} de menor perímetro el número de estaciones aumenta a 7.

Existen polígonos para los cuales, dada una distancia d , el conjunto de estaciones de una solución óptima al problema MinEstaciones yace sobre la d -envoltura más pequeña que los encierra. Un ejemplo de esto se muestra en la Figura 2.1. Sin embargo, este no siempre es el caso: en la Figura 2.3 se muestra un ejemplo en el que al colocar estaciones en cualquier posición distinta a los puntos mostrados se incrementará el número de estaciones necesarias; en este ejemplo, la estación representada por el punto x no está colocada sobre la d -envoltura más pequeña que encierra a \mathcal{P} . Otro ejemplo de esto es el polígono en forma de estrella de la Figura 2.4, en el que

la distancia entre vértices adyacentes del cierre convexo es $\frac{d}{2} + \epsilon$ para un ϵ arbitrariamente pequeño. En esta construcción, el número de estaciones necesarias cuando se restringen a estar sobre la d -envoltura más corta es casi el doble que el requerido en una solución óptima al problema MinEstaciones sin esta restricción.

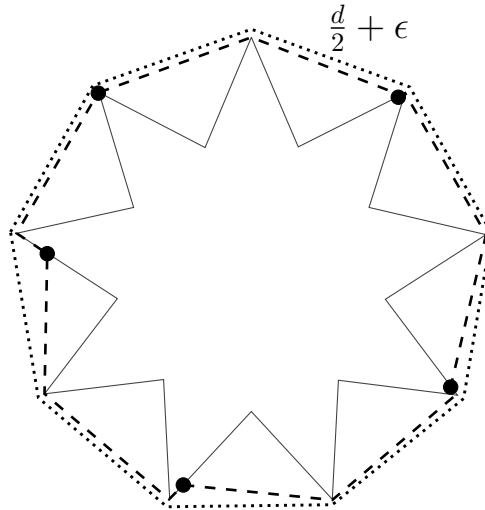


Figura 2.4: Un polígono \mathcal{P} tal que cualquier solución en que las estaciones de recarga se coloquen sobre la d -envoltura de menor perímetro (mostrada en líneas punteadas) requiere aproximadamente el doble de estaciones que una solución óptima. Se muestra la d -envoltura de la solución óptima en líneas discontinuas.

Observemos que en cualquier solución óptima para el problema MinEstaciones, las estaciones de recarga siempre están colocadas sobre $\partial\mathcal{P}$, pero no necesariamente en vértices de \mathcal{P} o de \mathcal{C} .

2.3. Trabajo relacionado

El problema del mínimo número de estaciones está relacionado con el problema de la galería de arte y sus variantes [80, 100], en particular el problema de la *ruta del vigilante* (del inglés *Watchman Route problem*), en el que buscamos una trayectoria en el interior de un polígono \mathcal{P} tal que cada punto de \mathcal{P} sea visible desde algún punto de dicha trayectoria. En contraste con el problema de la galería de arte que es NP-difícil incluso para familias restringidas de polígonos, ver [98], se han presentado varios algoritmos de complejidad temporal polinomial para el problema de la ruta del vigilante. Los primeros algoritmos para este problema fueron presentados con

complejidades desde $O(n^4 \log n)$ hasta $O(n^2)$, ver [25, 95, 18]. Sin embargo, en [61] se mostró que los algoritmos existentes podían *en realidad* tomar tiempo exponencial bajo ciertas condiciones y se intentó corregir el error mediante un nuevo algoritmo de tiempo $O(n^2)$. Dos años después se mostró que la corrección dada en [61] tenía el mismo defecto que los algoritmos anteriores, ver [96]; en este artículo se dió una nueva corrección en la forma de un algoritmo de tiempo $O(n^4)$. Para todos los algoritmos mencionados anteriormente se tiene la suposición de que la ruta del vigilante debe pasar por un punto p en el polígono, es decir, la ruta del vigilante tiene un punto inicial fijo. La variante de este problema en la que no se da un punto inicial para la ruta también ha sido estudiada y existe una solución de tiempo $O(n^5)$ según se demostró en [94].

Una versión del problema de la ruta del vigilante, que es más cercana al problema del mínimo número de estaciones, es estudiada en [79]. En esa variante se busca una ruta del vigilante contenida en el exterior del polígono. En este artículo se da un algoritmo de tiempo $O(n^4 \log n)$ para polígonos simples. Este algoritmo se basa en reducir el problema de vigilar el exterior en el problema de vigilar el interior mediante la transformación del polígono dado y utiliza como subrutina cualquier algoritmo que permita obtener la ruta del vigilante en el interior de un polígono. Dado que la versión original del algoritmo de la ruta externa utiliza como paso intermedio un algoritmo de tiempo $O(n^4)$ para obtener la ruta en el interior, podemos utilizar el algoritmo de la corrección dada en [96] y obtener una complejidad temporal de $O(n^4 \log n)$ en el algoritmo para encontrar la ruta externa.

La instalación de un conjunto de estaciones de recarga que satisfagan la restricción de rango de vuelo de los drones está relacionada con los problemas del área de *localización de instalaciones* (del inglés *facility location problems*), donde se busca encontrar las mejores ubicaciones posibles para colocar un conjunto de instalaciones que dan cierto tipo de servicio (por ejemplo, centros de distribución, farmacias o estaciones de bomberos), así como encontrar rutas óptimas para visitarlas. En [105] se estudió un problema de localización de estaciones y planeación de trayectorias en el que los autores formularon el problema utilizando programación lineal entera. Un ejemplo más reciente de este tipo de problemas en el que se dan soluciones heurísticas puede encontrarse en [71].

Otra área de estudio relacionada con el problema del mínimo número de estaciones es la de *redes de sensores inalámbricas*. En [68] los autores estudiaron el problema de desplegar un conjunto de sensores en una región que rodea un castillo de modo que cualquier intruso sea detectado por al menos k de estos sensores para una k dada. Una variante de este problema es estudiada en [14], donde el problema es

determinar la manera óptima en la que los sensores se deben mover para prevenir futuras intrusiones una vez que se ha detectado a un intruso.

2.4. Resultados preliminares

Sea \mathcal{P} un polígono con n vértices orientado en dirección horaria. Para poder comparar la posición de cualquier par de puntos de $\partial\mathcal{P}$ respecto a la primera estación, dada como un punto fijo $s \in \partial\mathcal{P}$, definimos un orden total (\mathcal{P}, \preceq) en los puntos de $\partial\mathcal{P}$ como sigue:

1. para cualquier punto $a \in \partial\mathcal{P}$, $s \preceq a$, y
2. para cualquier par de puntos $a, b \in \partial\mathcal{P}$ distintos de s , $a \preceq b$ si $[s, a] \subseteq [s, b]$.

Por conveniencia, agregaremos al orden un elemento s' que comparte con s su posición en $\partial\mathcal{P}$, tal que para cualquier $a \in \partial\mathcal{P}$, $a \preceq s'$; es decir, s y s' son, respectivamente, el mínimo y el máximo elemento de (\mathcal{P}, \preceq) . En adelante nos referiremos a (\mathcal{P}, \preceq) simplemente como \preceq .

Consideremos un punto $s \in \partial\mathcal{P}$ y el orden \preceq que define en los puntos de $\partial\mathcal{P}$. Definimos la distancia δ_d entre los puntos de $\partial\mathcal{P}$ como sigue:

1. $\delta_d(a, a) = 0$.
2. $\delta_d(a, b) = 1$ si $a \preceq b \in \mathcal{P}$ y existe una d -trayectoria de longitud a lo más d de a a b .
3. $\delta_d(a, b) = k$ si k es el entero más pequeño tal que existe una secuencia de puntos $p_0 = a, \dots, p_k = b$ que cumplen $\delta_d(p_i, p_{i+1}) = 1$, $i = 0, \dots, k - 1$.

Intuitivamente, la distancia $\delta_d(a, b)$ es el mínimo número de estaciones necesarias para llegar de a a b , contando a a como una de las estaciones.

La solución propuesta para el problema MinEstaciones está basada en el siguiente lema:

Lema 2.4.1. *Sean $w, x, y, z \in \partial\mathcal{P}$ tales que $w \preceq x \preceq y \preceq z$ en $\partial\mathcal{P}$, para los que se cumplen $\delta_d(w, y) \leq 1$ y $\delta_d(x, z) \leq 1$. Entonces, $\delta_d(w, z) \leq 2$, $\delta_d(w, x) \leq 2$ y $\delta_d(y, z) \leq 2$.*

Demostración. Supongamos que $\delta_d(w, z) > 1$, ya que de otro modo el lema es trivialmente cierto. Sean $\pi_{w,y}$ y $\pi_{x,z}$ las d -trayectorias más cortas que unen a w con y

y a x con z , respectivamente. Como $w \preceq x \preceq y \preceq z$, $\pi_{w,y}$ y $\pi_{x,z}$ se intersecan en un punto p . Se muestra un ejemplo de esto en la Figura 2.5.

Sean $\delta(x, p)$ la longitud del intervalo de $\pi_{x,z}$ entre p y x , y $\delta(p, y)$ la longitud del intervalo de $\pi_{w,y}$ entre p y y . Si $\delta(x, p) \leq \delta(p, y)$, entonces $\delta(w, p) + \delta(p, x) \leq 1$, y si $\delta(x, p) \geq \delta(p, y)$, entonces $\delta(y, p) + \delta(p, z) \leq 1$. Por lo tanto, $\delta_d(w, z) \leq 2$.

Las desigualdades $\delta_d(w, x) \leq 2$ y $\delta_d(y, z) \leq 2$ pueden ser demostradas con un argumento similar: si $\delta_d(w, x) > 2$ entonces $\delta(w, p) > 1$ o $\delta(x, p) > 1$, lo que contradice las hipótesis del lema. \square

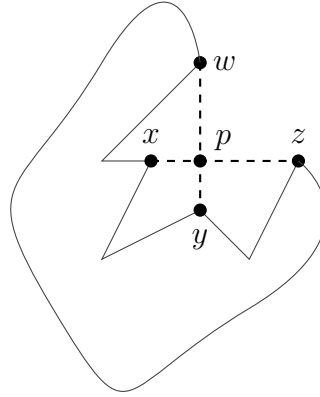


Figura 2.5: Ilustración del Lema 2.4.1.

Intuitivamente, el Lema 2.4.1 sugiere que en una solución óptima para el problema MinEstaciones cualquier dron volará sobre una curva cerrada \mathcal{C} que no se cruza a sí misma y que encierra a \mathcal{P} . Esta observación es formalizada en el siguiente lema. Denotamos como \mathcal{P}' al conjunto de puntos de $\partial\mathcal{P}$ en la frontera de $CH(\mathcal{P})$.

Lema 2.4.2. *Sea $s_0 \in \mathcal{P}'$ y sea $S = \{s_0, s_1, \dots, s_{k-1}\}$ una solución al problema MinEstaciones que avanza en dirección horaria alrededor de \mathcal{P} . Supongamos que S tiene la menor cantidad de estaciones posibles entre todas las soluciones que tienen a s_0 como punto inicial. Entonces $s_0 \preceq s_1 \preceq s_2 \preceq \dots \preceq s_{k-1}$.*

Demostración. Supongamos que todas las d -trayectorias π_i que unen a s_i con s_{i+1} son de longitud mínima. Sea p cualquier punto en \mathcal{P}' . Como $\mathcal{C} = \pi_0 \cup \dots \cup \pi_{k-1}$ encierra a \mathcal{P} , p está contenido en \mathcal{C} y es cubierto por \mathcal{C} exactamente una vez. Dado que $s_0 \in \mathcal{P}'$, lo anterior implica que s_0 es un extremo de π_0 , y que π_0 es una curva simple que avanza en dirección horaria a lo largo de \mathcal{C} .

Ahora, para llegar a una contradicción, supongamos que $s_i \preceq s_{i-1}$ para algún $i > 1$, y sea j el máximo valor tal que $s_j \preceq s_i$. Así, por la definición de S tenemos que $s_j \preceq s_i \preceq s_{j+1} \preceq s_{i-1}$, con $\delta_d(s_j, s_{j+1}) = 1$ y $\delta_d(s_i, s_{i-1}) = 1$. Por el Lema 2.4.1, $\delta_d(s_j, s_i) \leq 2$, y dado que S es una solución óptima, tenemos $s_{j+1} = s_{i-1}$.

Sea r el mínimo valor tal que $s_{i-1} \preceq s_r$. Entonces, tenemos que $s_i \preceq s_{r-1} \preceq s_{i-1} \preceq s_r$ y por lo tanto $s_i = s_{r-1}$. Como tenemos $s_{j+1} = s_{i-1}$ y $s_i = s_{r-1}$, se sigue $s_j \preceq s_i \preceq s_{i-1} \preceq s_r$, donde $\delta_d(s_j, s_{i-1}) = \delta_d(s_i, s_r) = 1$. Así, por el Lema 2.4.1, $\delta_d(s_j, s_r) \leq 2$, lo que contradice que S sea una solución óptima. Por lo tanto, $s_i \preceq s_{i+1}$ para toda i .

□

Notemos que con un argumento similar a la prueba del lema anterior podemos demostrar que para cualquier solución óptima $S = \{s_0, s_1, \dots, s_{k-1}\}$ para el problema MinEstaciones sin la restricción de que s_0 esté fijado, la d -envoltura \mathcal{C} resultante es una curva cerrada simple.

2.5. Algoritmo

A continuación, presentamos un algoritmo que construye una solución óptima para el problema MinEstaciones con la restricción de que un punto $v \in \mathcal{P}'$ esté contenido en el conjunto de estaciones devuelto.

Algoritmo SOLOPT

Entrada. Un polígono \mathcal{P} , un punto $v \in \mathcal{P}'$ y un valor $d > 0$.

Salida. Las ubicaciones de las estaciones en una solución óptima o casi óptima para MinEstaciones.

- 1 Sean $s_0 = s'_0 = y_{-1} = v$ y $y_0 = \max\{y : \delta_d(s_0, y) = 1\}$
- 2 Asignamos $S_0 = \{s_0\}$ e $i = 0$
- 3 **repetir**
- 4 $i = i + 1$
- 5 $y_i = \max\{y : \delta_d(s_{i-1}, y) = 2\}$
- 6 $s_i = \text{cualquier } s \in \{w : \delta_d(s_{i-1}, w) = 1 \text{ y } \delta_d(w, y_i) = 1\}$
- 7 Asignar $S_i = S_{i-1} \cup \{s_i\}$
- 8 **hasta** $y_i = s'_0$
- 9 **regresar** el último conjunto generado $S = S_i$

Afirmamos que este algoritmo regresa siempre una solución óptima o casi óptima

para MinEstaciones sin la restricción de que v esté en el conjunto S .

Teorema 2.5.1. *Dado un punto inicial $s_0 \in \mathcal{P}'$, si k es el menor valor tal que $s_k = s'_0$, entonces el conjunto de puntos $S = \{s_0, \dots, s_{k-1}\}$ regresado por el algoritmo SOLOPT es una solución óptima para el problema MinEstaciones con el requerimiento adicional de que una estación de recarga sea colocada en s_0 .*

Demostración. Notemos que si el perímetro de $CH(\mathcal{P})$ es menor o igual a 1, entonces cualquier estación de recarga en \mathcal{P}' es una solución óptima para el problema MinEstaciones. En el resto de la prueba supondremos que cualquier solución óptima contiene al menos dos estaciones de recarga.

Supongamos que $Z = \{z_0, \dots, z_{n-1}\}$ es una solución óptima para el problema MinEstaciones con una estación de recarga colocada en $s_0 = z_0$. A continuación, demostramos que $n = k$.

Por el Lema 2.4.2, podemos suponer que $v = z_0 \preceq z_1 \preceq \dots \preceq z_{n-1}$, que para toda i , $\delta_d(z_i, z_{i+1}) = 1$, y que $\delta_d(z_{n-1}, s'_0) = 1$. Consideremos el conjunto $S = \{s_0 = v, \dots, s_{k-1}\}$ regresado por el algoritmo SOLOPT y el conjunto $Y_{k-1} = \{y_{-1}, y_0, \dots, y_{k-1}\}$ de los valores y_i computados en la línea 5 de SOLOPT. Recordemos que $y_{k-1} = s_0$. Demostraremos por inducción que para todo i , $z_{i+1} \preceq y_i$, lo que implica que S es una solución óptima.

Esto se cumple para $i = 0$ ya que $z_1 \preceq y_0$ por definición. Supongamos que $z_i \preceq y_{i-1}$. Mostraremos que $z_{i+1} \preceq y_i$. Sea j el valor mínimo tal que $z_i \preceq y_{j-1}$. Entonces $j \leq i$, y como $v = y_{-1} \prec z_i$ tenemos que $j \geq 1$. Como j es mínimo, sabemos que $y_{j-2} \preceq z_i \preceq y_{j-1}$, y por definición de y_{j-2} , sabemos que $s_{j-1} \preceq y_{j-2}$. Al combinarlos, tenemos que $s_{j-1} \preceq z_i \preceq y_{j-1}$. Entonces, como por construcción $\delta_d(s_{j-1}, y_{j-1}) = 1$, el Lema 2.4.1 implica que $z_{i+1} \preceq y_j \preceq y_i$. Con esto termina el paso inductivo.

Por lo tanto, el mínimo k tal que $\delta_d(s_k, s'_0) = 1$ y el mínimo n tal que $\delta_d(y_n, s'_0) = 1$ son el mismo valor, y en consecuencia, $S = \{s_0 = v, \dots, s_{k-1}\}$ es una solución óptima para el problema MinEstaciones con el requerimiento adicional de que una estación de recarga sea colocada en s_0 . \square

Si consideramos el problema MinEstaciones sin el requerimiento de que un punto fijo s_0 pertenezca al conjunto de estaciones de recarga tenemos el siguiente resultado:

Teorema 2.5.2. *Sea $s_0 = v \in \mathcal{P}'$. El conjunto $S = \{s_0, \dots, s_{k-1}\}$ que regresa el algoritmo SOLOPT es una solución óptima o casi óptima para el problema MinEstaciones.*

Demostración. Supongamos que $Z = \{z_0, \dots, z_{n-1}\}$ es una solución óptima al problema MinEstaciones y que $S = \{s_0, \dots, s_{k-1}\}$ es la solución que regresa el algoritmo SOLOPT. Ahora probamos que $k = n$ o $k = n + 1$.

Dado que s_0 está en \mathcal{P}' existe una d -trayectoria más corta entre algún par z_i, z_{i+1} que contiene a s_0 . Por lo tanto, al agregar a s_0 a $Z = \{z_0, \dots, z_{n-1}\}$ obtenemos una solución óptima o casi óptima al problema MinEstaciones. \square

Notemos que, con modificaciones menores, el algoritmo SOLOPT puede ser utilizado para resolver el problema MinEstaciones para una trayectoria poligonal en lugar de un polígono simple. Esto podría ser útil para patrullar la costa de una región que no esté totalmente rodeada de agua.

2.5.1. Análisis de complejidad

En esta sección demostramos que el algoritmo SOLOPT se puede implementar de manera que se ejecute en tiempo $O(n^2 + Ln)$, donde L es el perímetro de \mathcal{P} dividido entre d . En adelante supondremos que s_0 es un punto en \mathcal{P}' .

El algoritmo SOLOPT debe resolver repetidamente el siguiente problema al cual llamamos *problema de los dos saltos*. Dado un punto s_i , queremos encontrar un punto $y_{i+1} = \max\{y : \delta_d(s_i, y) = 2\}$ con respecto a \preceq , y un punto $s_{i+1} \in \{w : \delta_d(s_i, w) = 1 \wedge \delta_d(w, y_{i+1}) = 1\}$; en la Figura 2.6 se muestra un ejemplo. Vamos a demostrar que, después de realizar un preprocesamiento de tiempo $O(n^2)$ en \mathcal{P} , podemos resolver el problema de los dos saltos en tiempo lineal para cada s_i .

Algunos puntos sobre la frontera de \mathcal{P} serán de particular interés para el computo de s_{i+1} a partir de s_i . Decimos que un punto x de una arista e es la *proyección* de un vértice p_i sobre e si $x \preceq p_i$ y el segmento de línea que los une es un d -puente de \mathcal{P} perpendicular a e . Ver la Figura 2.6a. Del mismo modo, decimos que un punto x de una arista e es la *d -proyección* de una arista f sobre e si existe un punto $y \in f$ tal que el segmento de línea que une a x con y es un d -puente de \mathcal{P} perpendicular a e . Ver la Figura 2.6b.

Lema 2.5.3. *Dado s_i , el punto s_{i+1} es un vértice de \mathcal{P} , la proyección de un vértice sobre una arista, la d -proyección de una arista sobre otra, o un punto tal que*

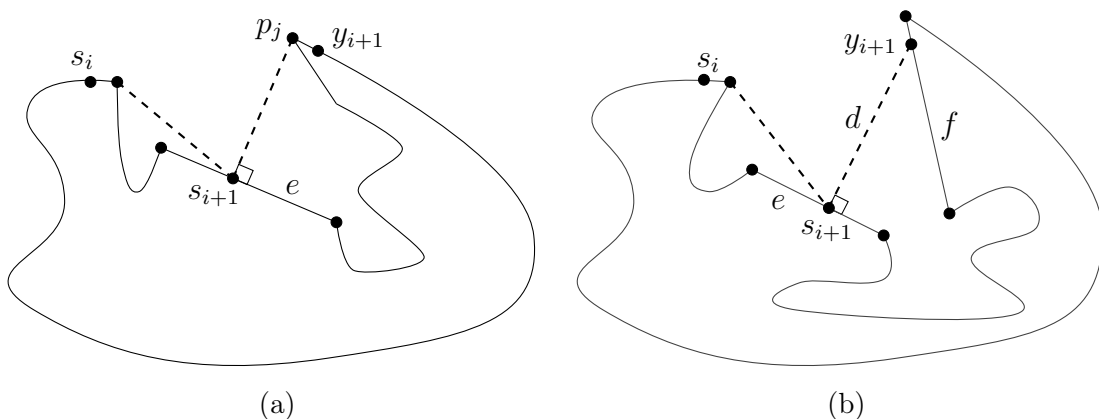


Figura 2.6: El problema de los dos saltos. (a) s_{i+1} es la proyección del vértice p_j sobre la arista e . (b) s_{i+1} es una d -proyección de la arista f sobre la arista e .

$$\delta(s_i, s_{i+1}) = d.$$

Demostración. Supongamos que el punto s_{i+1} no es un vértice de \mathcal{P} y que $\delta(s_i, s_{i+1}) < d$. Sea e la arista de \mathcal{P} que contiene a s_{i+1} , ver Figura 2.6. Notemos que, por la elección de y_{i+1} , $\delta(s_{i+1}, y_{i+1}) = d$. Si s_{i+1} no es la proyección de un vértice en e ni la d -proyección de una arista en e , entonces puede ser movido sobre la arista e y así hacer que y_{i+1} se mueva hacia delante, lo que contradice la definición de y_{i+1} . \square

Notemos que puede existir un número lineal de puntos a distancia d de una estación s_i . Sin embargo, entre todos ellos solamente necesitamos considerar el máximo con respecto a \preceq como un candidato para colocar s_{i+1} . Probamos esto a continuación.

Lema 2.5.4. *Sean w, x, y, z puntos en $\partial\mathcal{P}$ tales que $w \preceq x \preceq y \preceq z$, $\delta(w, x) = \delta(w, y) = d$, $\delta(w, z) > d$, y $\delta(x, z) = \ell$. Entonces, $\delta(y, z) \leq \ell$.*

Demostración. Sean $\pi_{w,y}$ y $\pi_{x,z}$, respectivamente, las d -trayectorias más cortas de w a y y de x a z . Sea r un punto de intersección de $\pi_{w,y}$ y $\pi_{x,z}$. Notemos que r siempre existe por la elección de los cuatro puntos en $\partial\mathcal{P}$. Sean $\delta(w, r)$ y $\delta(r, y)$ las distancias a lo largo de $\pi_{w,y}$ entre w y r , y entre r y y , respectivamente. Definimos $\delta(x, r)$ y $\delta(r, z)$ de forma análoga, pero a lo largo de $\pi_{x,z}$. Como $\delta(w, z) > d$, tenemos que $\delta(r, y) < \delta(r, z)$. Ahora supongamos que $\delta(r, y) > \delta(x, r)$. Esto implica que $\delta(w, r) + \delta(r, x) < d$, lo que es una contradicción a la suposición de que $\delta(w, x) = d$. Por lo tanto, $\delta(r, y) \leq \delta(x, r)$ y $\delta(y, z) \leq \ell$. \square

Si bien puede existir un número cuadrático de proyecciones de vértices y d -proyecciones de aristas, afirmamos que solo es necesario considerar un número lineal de ellas como puntos candidatos para obtener todos los s_i .

Sean e y f aristas de \mathcal{P} . Decimos que $e \triangleleft f$ si para cualquier punto x en el interior de e y cualquier punto y en el interior de f , $x \preceq y$.

Lema 2.5.5. *Para cualquier arista e de \mathcal{P} necesitamos almacenar únicamente tres puntos:*

1. *La d -proyección mínima (con respecto a \preceq) sobre e de una arista e' tal que $e \triangleleft e'$.*
2. *El extremo que no está en e del puente que genera la d -proyección máxima (con respecto a \preceq) sobre e de una arista e' tal que $e' \triangleleft e$. En este caso, el punto almacenado está en e' .*
3. *La proyección mínima (con respecto a \preceq) de un vértice en e .*

Demostración. Caso 1. Sean x y x' d -proyecciones sobre e de dos aristas distintas f y f' , respectivamente, tales que $e \triangleleft f$ y $e \triangleleft f'$. Sea $s(x, y)$ el d -puente perpendicular a e que contiene a x como un extremo, i.e., $y \in f$ y $s(x, y)$ tiene longitud d . Definimos a $s(x', y')$ de forma análoga. Debido a la longitud de $s(x, y)$ (respectivamente, de $s(x', y')$), si colocamos una estación de recarga en x (respectivamente, en x') entonces también tenemos que colocar una estación de recarga en y (respectivamente, en y'). Supongamos sin pérdida de generalidad que $x \preceq x'$, ver la Figura 2.7a. Dado que todos los puentes que definen d -proyecciones de aristas sobre e son paralelos, esto implica que $f' \triangleleft f$ y $y' \preceq y$. Además, como el intervalo $[x, y]$ contiene al intervalo $[x', y']$, al colocar una estación de recarga en x garantizamos que ambos intervalos están protegidos. Por lo tanto, al elegir como s_{i+1} a la mínima d proyección de una arista sobre e maximizamos a y_{i+1} con respecto a \preceq .

Caso 2. Este caso es análogo al caso anterior, ver Figura 2.7b.

Caso 3. Sean x y x' las proyecciones de dos vértices distintos, p_i y p_j , respectivamente, sobre una arista e . Sean $s(x, p_i)$ y $s(x', p_j)$ los d -puentes correspondientes a dichas proyecciones. Supongamos sin pérdida de generalidad que $x \preceq x'$. Esto implica que $p_j \preceq p_i$ y que al colocar una estación de carga en x garantizamos que los intervalos $[x, p_i]$ y $[x', p_j]$ estén protegidos. Se muestra un ejemplo en la Figura 2.8a. Mostraremos que al colocar una estación en x un dron puede alcanzar un punto mayor en $\partial\mathcal{P}$ con respecto a \preceq . Sea $w \in P$ un punto tal que $p_i \preceq w$ y sean $\pi_{x,w}$ y $\pi_{x',w}$ las

d -trayectorias más cortas que unen a x con w y a x' con w , respectivamente. Sea r el punto de intersección de $s(x, p_i)$ y $\pi_{x', w}$. Notemos que los puntos x , x' y r forman un triángulo rectángulo cuyo ángulo recto está en x , ver Figura 2.8a. Por lo tanto, la longitud de $\pi_{x, w}$ es menor que la longitud de $\pi_{x', w}$ y podemos obtener un y_{i+1} mayor al elegir la mínima proyección de un vértice en e como s_{i+1} . \square

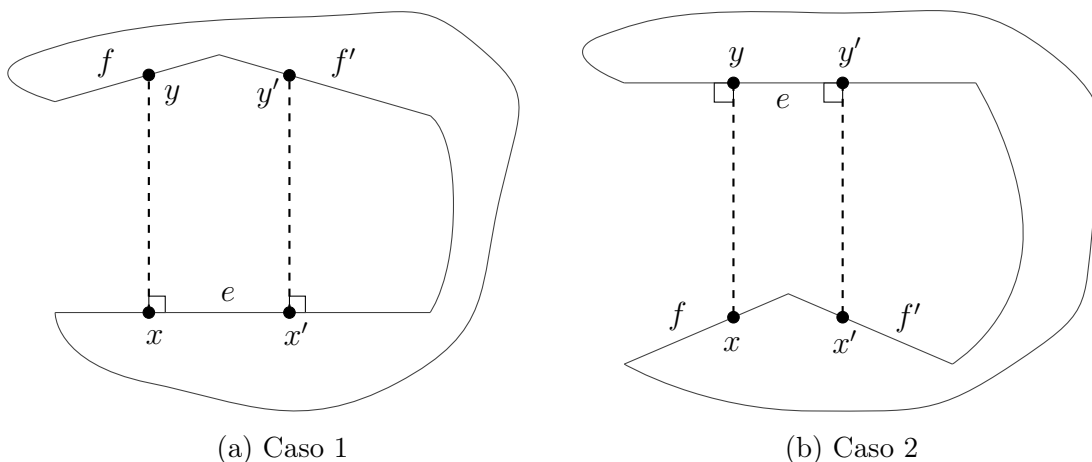


Figura 2.7: Ilustración de los casos 1 y 2 del Lema 2.5.5. En ambos casos es necesario almacenar únicamente el punto x .

Como un paso intermedio para computar los puntos candidatos sobre $\partial\mathcal{P}$ obtenemos, para cada arista e , el conjunto que contiene cada punto $x \in \partial\mathcal{P}$ para los que existe un segmento perpendicular a e que une a x con e y está completamente contenido en $Ext(\mathcal{P})$. Decimos que x es *ortogonalmente visible* desde e .

Definimos una *tapa* de \mathcal{P} como una arista de $CH(\mathcal{P})$ que no es una arista de \mathcal{P} . Cada tapa $h = s(a, b)$ define un polígono \mathcal{P}_h cuya frontera es la unión de h y el intervalo entre a y b cuya intersección con $CH(\mathcal{P})$ son solo los puntos a y b . Notemos que cualquier proyección de un vértice o d -proyección de una arista está definida por un puente cuyos dos extremos están contenidos en el mismo polígono \mathcal{P}_h , ya que de no ser así dicho puente intersectaría a $Int(\mathcal{P})$. Por lo anterior, solo necesitamos obtener los conjuntos de puntos ortogonalmente visibles desde aquellas aristas contenidas en algún polígono \mathcal{P}_h , los cuales estarán contenidos en el mismo \mathcal{P}_h .

Para el siguiente lema vamos a asumir que ya hemos computado los polígonos correspondientes a todas las tapas de \mathcal{P} , así como la triangulación de cada polígono. Esto puede realizarse en tiempo total $O(n)$ siguiendo los resultados en [76] y [22].

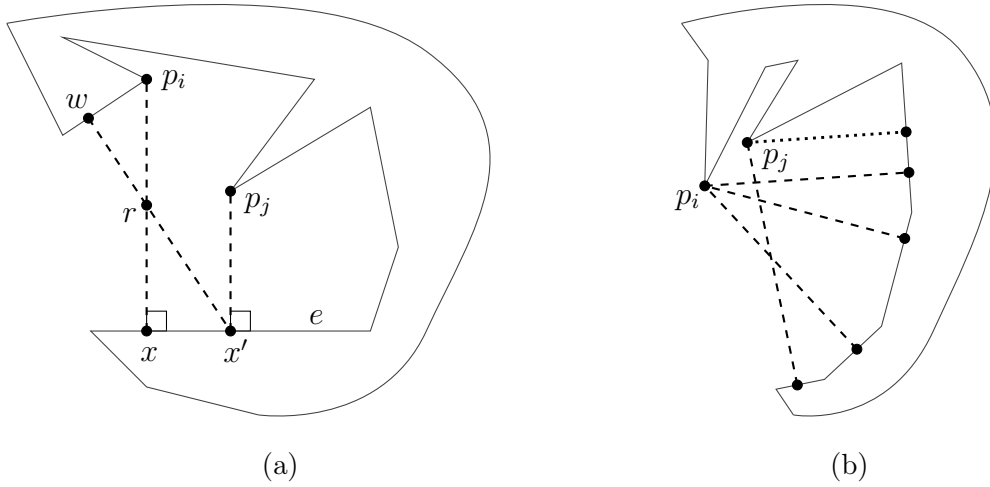


Figura 2.8: Ilustración del caso 3 del Lema 2.5.5. (a) La trayectoria más corta de x a w tiene menor longitud que la de x' a w . (b) Necesitamos considerar todas la proyecciones de vértices mostradas excepto la que es extremo del segmento punteado.

Lema 2.5.6. *Podemos encontrar el conjunto de todos los segmentos de \mathcal{P} que son ortogonalmente visibles desde cualquier arista de \mathcal{P} en tiempo $O(n)$. Además, este conjunto tiene tamaño $O(n)$.*

Demostración. Sea $h = s(a, b)$ una tapa de \mathcal{P} y sea $e = s(u, v)$ una arista de \mathcal{P}_h distinta a h .

Primero obtenemos el conjunto de puntos de \mathcal{P}_h visibles desde algún punto en e . Este conjunto, el cual denotamos con $VP(\mathcal{P}_h, e)$, puede ser obtenido en tiempo $O(n)$, ver [60].

Supongamos sin pérdida de generalidad que $u \prec v$. Sea R la región contenida entre las líneas perpendiculares a e que pasan por u y por v , y a la derecha de la línea dirigida de u a v . Es fácil ver que cualquier punto ortogonalmente visible desde e debe estar contenido en el polígono $\mathcal{R}_e = VP(\mathcal{P}_h, e) \cap R$, el cual puede ser obtenido en tiempo $O(n)$ al intersectar $VP(\mathcal{P}_h, e)$ con ambas líneas que definen a R . En adelante, supondremos sin pérdida de generalidad que e es horizontal y que el interior de \mathcal{R}_e está arriba de e .

Notemos que los ángulos internos de \mathcal{R}_e en ambos vértices de $e = s(u, v)$ son menores a 180 grados, por lo que podemos utilizar el siguiente resultado de Ghosh et

al. [58]: para cualquier vértice p en \mathcal{R}_e , la trayectoria más corta de u a p , denotada como $\rho_{u,p}$, da vuelta a la izquierda en cada vértice de la trayectoria, mientras que $\rho_{v,p}$ dobla a la derecha en cada vértice de dicha trayectoria.

Decimos que un vértice es un *vértice de giro* si el segmento vertical maximal que contiene a p y está completamente contenido en \mathcal{R}_e divide a \mathcal{R}_e en tres subpolígonos; se muestra un ejemplo de esto en la Figura 2.9a. Si dos de estos subpolígonos están a la derecha (izquierda) de $s(x, y)$ decimos que p es un vértice de giro *derecho* (*izquierdo*). Sea x el extremo superior de $s(x, y)$. El segmento $s(p, x)$ separa a \mathcal{R}_e en dos subpolígonos, uno de los cuales contiene a e . Denotamos con $\mathcal{R}_e(p)$ al subpolígono generado por $s(p, x)$ que no contiene a e . Notemos que cualquier punto en \mathcal{R}_e que no sea ortogonalmente visible desde e está contenido en el subpolígono $\mathcal{R}_e(p)$ para algún vértice de giro p , además, ningún punto en $\mathcal{R}_e(p) \setminus s(p, x)$ es ortogonalmente visible desde e .

Sea p un vértice de giro de \mathcal{R}_e y sea x el extremo superior del segmento vertical maximal completamente contenido en \mathcal{R}_e que contiene a p . Afirmamos que la línea vertical que pasa por p , ℓ_p , no interseca a ningún punto de $\mathcal{R}_e(p) \setminus s(p, x)$. Supongamos que, por el contrario, existe un punto x' de $\mathcal{R}_e(p) \setminus s(p, x)$ contenido en ℓ_p . Entonces, existe un vértice q en $\mathcal{R}_e(p) \setminus s(p, x)$ tal que $\rho_{v,x'}$ dobla a la izquierda en q o $\rho_{u,x'}$ dobla a la derecha en q , lo cual es una contradicción al resultado de Ghosh [58]. Se muestra un ejemplo en la Figura 2.9b. Se sigue $\mathcal{R}_e(p) \cap \ell_p = s(p, x)$, lo que sugiere el siguiente algoritmo para remover de \mathcal{R}_e el subpolígono $\mathcal{R}_e(p)$ por cada vértice de giro p .

Para remover los subpolígonos generados por vértices de giro derechos recorremos las aristas de \mathcal{R}_e en sentido horario desde v hasta u . Asignamos una variable booleana *aristaVisible* como *verdadera*. Sea $f = s(q, r)$, $q \prec r$, la arista actual en el recorrido.

- Si *aristaVisible* es *verdadera* revisamos si r es un vértice de giro derecho. En el caso afirmativo, asignamos *aristaVisible* como *falsa* y almacenamos la arista vertical que contiene a r , denotada ℓ_r , y la arista f .
- Si *aristaVisible* es *falsa*, entonces previamente habíamos almacenado la última arista visible $g = s(o, p)$, donde p es un vértice de giro derecho, así como la línea vertical por p , ℓ_p . Revisamos si $x = f \cap \ell_p$ es no vacío. Si es el caso, reemplazamos el intervalo $[p, x]$ de \mathcal{R}_e con el segmento vertical $s(p, x)$, asignamos *aristaVisible* como *verdadera*, y descartamos g y ℓ_p .

Análogamente, removemos los subpolígonos definidos por vértices de giro izquierdos recorriendo \mathcal{R}_e en sentido antihorario desde u hasta v . Como cada arista de \mathcal{R}_e

es visitado a lo más dos veces, la remoción de los subpolígonos definidos por todos los vértices de giro toma tiempo $O(n)$. Denotamos por \mathcal{R}'_e al polígono obtenido después de ambos recorridos.

Para obtener el conjunto de puntos de \mathcal{P} ortogonalmente visibles desde e , solo tenemos que descartar de $\mathcal{R}'_e(p)$ a e , a cualquier segmento contenido en la tapa de \mathcal{P}_h , y a los segmentos verticales agregados en el proceso anterior.

Dado que \mathcal{P}_h no tiene hoyos, cada arista de \mathcal{P}_h aporta a lo más un segmento a \mathcal{R}'_e . Por lo tanto, el conjunto de segmentos de \mathcal{P} ortogonalmente visibles desde la arista e tiene tamaño $O(n)$. \square

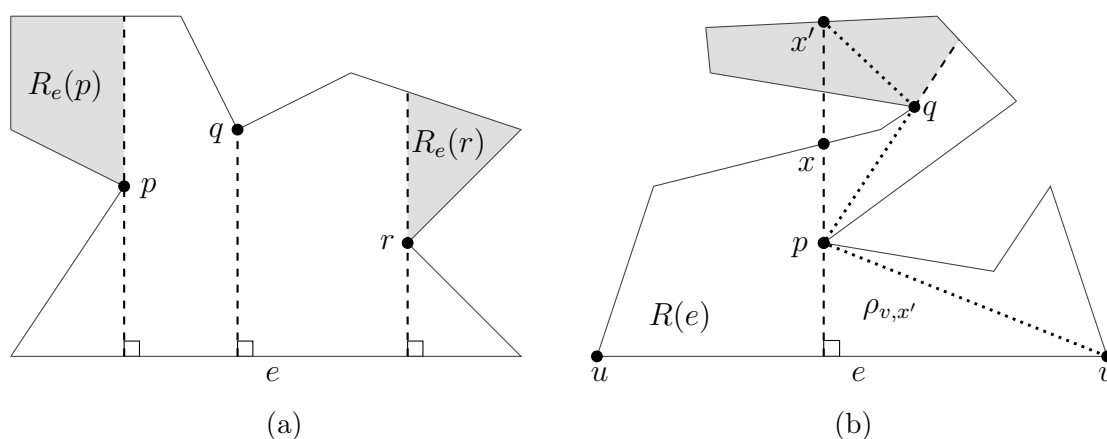


Figura 2.9: (a) p es un vértice de giro izquierdo, r es un vértice de giro derecho, y q no es un vértice de giro. (b) Ni x' ni cualquier punto en la región sombreada está en R_e : la trayectoria más corta de v a x' gira a la izquierda en q .

Lema 2.5.7. *Podemos encontrar las proyecciones descritas en el Lema 2.5.5 en tiempo $O(n)$ para cualquier arista e .*

Demostración. Supongamos que $e = s(u, v)$, $u \preceq v$. Por el Lema 2.5.6, podemos encontrar el conjunto W de todos los vértices y segmentos de aristas de \mathcal{P} ortogonalmente visibles desde e en tiempo $O(n)$; además, W tiene tamaño lineal. Sea W_B el subconjunto de elementos de W tal que los puntos que los conforman son menores que v con respecto a \preceq . Similarmente, sea W_A el subconjunto de elementos de W tal que los puntos que los conforman son mayores que v con respecto a \preceq .

Las d -proyecciones correspondientes a los primeros dos casos del Lema 2.5.5 son obtenidas como sigue. Sea ℓ la línea paralela a e ; a la izquierda de la línea dirigida de

u a v , y a distancia d de e . Primero computamos la intersección de ℓ con W_B y W_A , lo cual toma tiempo $O(n)$ por el tamaño de W . Para obtener el punto descrito en el Caso 1 de la prueba del Lema 2.5.5, tomamos el punto de $\ell \cap W_A$ que sea máximo con respecto a \preceq y almacenamos el punto de intersección de e con la línea que contiene a q y es perpendicular a e . Para obtener el punto descrito en el Caso 2 de la prueba del Lema 2.5.5 almacenamos el mínimo punto en $\ell \cap W_B$.

La proyección de un vértice sobre e descrita en el Caso 3 del Lema 2.5.5 es obtenida como sigue. Para cada vértice de \mathcal{P} en W_B computamos su distancia con respecto a e y después almacenamos el máximo con respecto a \preceq de aquellos vértices cuya distancia a e sea a lo más e . \square

De los puntos candidatos listados en el Lema 2.5.3 nos resta por mostrar la manera de computar los puntos a distancia exactamente d de la última estación colocada. Es decir que, dado un punto $x \in \mathcal{P}$, necesitamos encontrar el máximo w , $x \preceq w$, tal que $\delta(x, w) = d$. Guibas et al. [60] probaron que, dada la triangulación de un polígono \mathcal{R} y un punto $p \in \mathcal{R}$, las trayectorias más cortas (según la distancia euclidiana) desde p hacia todos los vértices de \mathcal{R} pueden ser encontrados en tiempo lineal (ver también [70]). La unión de todas las trayectorias más cortas desde un punto p a los vértices de \mathcal{R} es un árbol llamado *árbol de rutas más cortas* de \mathcal{R} con respecto a p .

Sea \mathcal{R} el polígono obtenido al encerrar a \mathcal{P} en un rectángulo suficientemente grande y conectar uno de los lados del rectángulo al punto de inicio del polígono, x_0 , por medio de un corredor delgado. El polígono \mathcal{R} puede ser obtenido en tiempo $O(n)$, ver [79]. Notemos que \mathcal{R} tiene $m \leq n + 8$ vértices y \mathcal{P} está contenido en el exterior de \mathcal{R} . Asignamos a los puntos de \mathcal{R} que son también puntos en \mathcal{P} el mismo orden que en \mathcal{P} .

En adelante supondremos ya hemos obtenido a \mathcal{R} junto con su triangulación.

Lema 2.5.8. *Dado cualquier punto $x \in \mathcal{P}$, el punto $w \in \mathcal{P}$ con $\delta(x, w) = d$ tal que w es máximo con respecto a \preceq puede ser encontrado en tiempo $O(n)$.*

Demostración. Sea x un punto en \mathcal{P} y sea x' su punto correspondiente en \mathcal{R} . Computamos la trayectoria más corta $\rho(x', y)$ de x' a cada vértice $y \in \mathcal{R}$ tal que y es también un vértice de \mathcal{P} y se cumple $x' \prec y$. Sea T el árbol de trayectorias más cortas obtenido por la unión de todas estas trayectorias. Sea M el conjunto de vértices de T tal que para cualquier $w \in M$, $\delta(x', w) \leq d$, y tal que w comparte un vértice de \mathcal{R} con un vértice y tal que $\delta(x', y) > d$. El conjunto M puede ser encontrado en tiempo $O(n)$ al recorrer T comenzando desde su raíz x' .

Observemos que cualquier punto de \mathcal{R} a distancia d de x' cumple con ser uno de los siguientes tipos de puntos:

- Un elemento de M .
- Un punto en una arista $e = s(u, v)$ de \mathcal{R} , $u \prec v$, tal que $e \in E(T)$. En este caso, $u \in M$ y $\delta(x', v) > d$.
- Un punto en una arista $e = s(u, v)$ de \mathcal{R} , $u \prec v$, tal que $e \notin E(T)$. Notemos que, en esta caso, $\delta(x', v) > d$. Además, existe exactamente un $z \in M$ tal que $(z, u) \in E(T)$.

Por lo tanto, para encontrar todos los puntos a distancia exactamente d de x' es suficiente considerar las aristas que tienen como extremo a un punto que en T es vecino de algún elemento en M . Dado que cada vértice es adyacente a lo más a un elemento de M , esto toma tiempo $O(n)$. En el último paso necesitamos encontrar el máximo entre todos los puntos a distancia d de x' , lo que toma tiempo $O(n)$. Por lo tanto, nuestra afirmación se cumple. \square

Uniendo todo lo anterior obtenemos el siguiente resultado.

Teorema 2.5.9. *Sea \mathcal{P} un polígono con n vértices y sea s_0 un punto en \mathcal{P}' . Entonces, SOLOPT regresa una solución óptima S al problema MinEstaciones tal que $s_0 \in S$ en tiempo $O(n^2 + \frac{L}{d}n)$, donde L es el perímetro de \mathcal{P} .*

Demostración. Por el Lema 2.5.3, dado s_i , el punto s_{i+1} es un punto en ∂P a distancia exactamente d de s_i , un vértice de \mathcal{P} , la proyección de un vértice en una arista, o la d -proyección de una arista sobre otra.

De entre los puntos a distancia d de s_i nos interesa solamente el máximo respecto a $O(n)$, como se muestra en el Lema 2.5.4. Por el Lema 2.5.8, este punto puede ser encontrado en tiempo $O(n)$

Si bien podrían existir $O(n^2)$ proyecciones de vértices y d -proyecciones de aristas, el Lema 2.5.8 nos dice que en el conjunto de puntos candidatos solamente necesitamos guardar a lo más tres proyecciones por cada arista de \mathcal{P} . Además, este conjunto puede ser encontrado en tiempo $O(n)$.

El conjunto de puntos candidatos para computar todos los elementos del conjunto S tiene tamaño $O(n)$. Para cada punto candidato x , computamos el máximo punto a distancia d de x y asociamos este punto a x , el cual por el Lema 2.5.8 toma tiempo $O(n)$ por candidato.

Es fácil ver que solamente necesitamos considerar los puntos candidatos contenidos en el intervalo de s_i al máximo punto a distancia d de s_i . De todos estos candidatos, elegimos como s_{i+1} aquel que maximice y_{i+1} , lo cual puede ser realizado en tiempo $O(n)$. Como podría ser necesario colocar $O(\frac{L}{d})$ estaciones de recarga, este paso toma tiempo $O(\frac{L}{d}n)$. Por lo tanto, el conjunto S puede ser encontrado en tiempo $O(n^2 + \frac{L}{d}n)$. \square

Capítulo 3

Cobertura de ortoárboles con balizas

3.1. Introducción

En este capítulo estudiamos una variante reciente del problema de la galería de arte conocida como el problema de *cobertura con balizas*. Una *baliza* (del inglés *beacon*) es un tipo de guardia que atrae a los puntos hacia sí mismo de forma similar a la de un imán que atrae partículas ferrosas. Cuando una baliza b atrae a un objeto colocado sobre un punto p en un polígono o poliedro \mathcal{P} , el objeto en p se mueve en línea recta hacia b hasta que alcanza a b o se encuentra con un obstáculo. El objeto puede deslizarse a lo largo de los obstáculos siempre y cuando disminuya de manera local su distancia hacia b al desplazarse. Se dice que una baliza b *cubre* a un punto p si un objeto colocado en p logra llegar hasta b al ser atraído por dicha baliza. Cuando esto no pasa se dice que el objeto en p se ha *atorado*, es decir, que dicho objeto llegó a un punto p' para el que existe un $\epsilon > 0$ tal que cualquier punto del poliedro a distancia a lo más ϵ de p' está a mayor distancia de b que p' . Se dice que un conjunto de balizas *cubre* un poliedro \mathcal{P} si todo punto de \mathcal{P} está cubierto por alguna baliza. En este capítulo estudiamos una familia de poliedros ortogonales llamados ortoárboles, los cuales tienen la propiedad de que la gráfica de adyacencias de su descomposición en componentes básicos es un árbol. El resultado obtenido es la demostración de que cualquier ortoárbol con n vértices puede ser cubierto por un conjunto de a lo más $\lfloor n/12 \rfloor$ balizas; esta cota es justa, como se probó en la tesis de maestría del autor.

3.2. Trabajo relacionado

Aún cuando el problema de la galería de arte es un problema clásico de la geometría computacional estudiado desde la década de 1970, no se conoce mucho sobre este problema en \mathbb{R}^3 . Esto sucede a causa de su dificultad, ya que por lo general no es posible utilizar las técnicas utilizadas para resolver las variantes del problema de la galería de arte en \mathbb{R}^2 . A diferencia de lo que ocurre en el plano, existen poliedros que no pueden ser triangulados, es decir, que no pueden descomponerse en tetraedros sin agregar nuevos vértices; un ejemplo de esto es el poliedro de Schönhardt [84], que es el poliedro no convexo más simple conocido que no se puede descomponer en tetraedros. Además, existen poliedros ortogonales que no pueden ser vigilados por guardias colocados en sus vértices, y esto sucede aún cuando se coloque un guardia en cada uno de esos vértices. Uno de los primeros ejemplos de poliedros que no pueden ser vigilados desde sus vértices es conocido como *poliedro de Seidel*, el cual contiene $O(n^2)$ vértices y requiere $O(n^3)$ guardias para ser vigilado [80]. Otro ejemplo de esto, obtenido mediante una construcción más sencilla es el poliedro conocido como *octoplex* [77]. Por lo anterior, ha sido necesario buscar nuevos tipos de guardia, como los guardias arista. Se sabe que cualquier poliedro puede ser iluminado por un conjunto de *guardias arista*, es decir, por un conjunto de aristas tales que cualquier punto en el interior del poliedro sea visible desde algún punto de alguna arista de este conjunto. En 1996 Urrutia [100] demostró que $\lfloor e/6 \rfloor$ guardias arista son siempre suficientes para vigilar cualquier poliedro ortogonal de e aristas y conjeturó que la cota justa es de $\lfloor e/12 \rfloor \pm c$. Intuitivamente, el *género* de un poliedro es la cantidad de hoyos que tiene dicho poliedro; por ejemplo, cualquier poliedro homeomorfo al toro tiene género uno. En 2011 Benbernou et al. [12] demostraron que cualquier poliedro ortogonal de género g y e aristas puede ser vigilado usando $\lfloor 11e/72 \rfloor - g/6 - 1$ guardias arista *cerrados*, es decir, guardias arista que incluyen a los vértices de las aristas. Recientemente Aldana et al. [3] probaron un resultado similar, pero utilizando un modelo de guardias arista más restringido: guardias arista que no incluyen los vértices y que tienen un ángulo de iluminación de $\pi/2$. También existen resultados para poliedros en los que los guardias son caras del poliedro ortogonal en lugar de vértices o de aristas, los cuales fueron presentados por Viglietta [103].

Los primeros resultados para el problema de cobertura con balizas en el plano fueron presentados en 2011 por Biro et al. [16], quienes mostraron que encontrar un conjunto mínimo de balizas que cubran un polígono simple es un problema *NP*-difícil. Biro [15] mostró también que $\lfloor n/3 \rfloor$ balizas son siempre suficientes y $\lfloor 4n/13 \rfloor$ balizas son a veces necesarias para cubrir un polígono simple de n vértices. En el caso de los polígonos ortogonales, Bae et al. [9] demostraron que $\lfloor n/6 \rfloor$ balizas son

siempre suficientes y a veces necesarias. También existen resultados para el problema de cobertura con balizas en \mathbb{R}^3 . Encontrar un conjunto mínimo de balizas que cubran un poliedro es *NP*-difícil, como lo demostró Cleve [28] en 2017. Además, tal como sucede para el problema de la galería de arte, existen poliedros ortogonales que no pueden ser cubiertos colocando balizas en sus vértices; un ejemplo de esto se puede consultar en Aldana et al. [5]. Lo anterior justifica el estudio del problema de cobertura con balizas en familias no triviales de ortoárboles que puedan ser cubiertos desde sus vértices, como hacemos en esta sección.

3.3. Definiciones

Un *poliedro ortogonal* en \mathbb{R}^3 es un poliedro tal que cada una de sus caras es perpendicular a uno de los tres ejes coordenados. Llamaremos *caja* a cualquier poliedro ortogonal con la forma de un prisma rectangular.

Un *ortoárbol* (ver [34]) \mathcal{P} es un poliedro ortogonal simplemente conexo el cual es la unión de un conjunto de cajas $S = \{B_1, \dots, B_n\}$ en \mathbb{R}^3 pegadas cara a cara de tal forma que la gráfica cuyos vértices son B_1, \dots, B_n , y en la que dos de ellos están unidos por una arista si comparten una cara, es un árbol. Denotaremos esta gráfica como $T_{\mathcal{P}}$. En la Figura 3.1 se muestra un ejemplo de un ortoárbol. Notemos que no todos los vértices de una caja $B_i \in S$ son vértices del ortoárbol \mathcal{P} . Cuando un vértice de una caja no es un vértice de \mathcal{P} se debe a que dicho vértice pertenece a exactamente dos cajas de S . En el resto de los casos posibles, cuando un vértice v pertenece a una, tres o cuatro cajas, v es siempre un vértice de \mathcal{P} .

Un par de cajas de \mathcal{P} que comparten una cara tal que ninguno de los vértices de dicha cara es vértice de \mathcal{P} es llamado un *par redundante* de cajas. Si esto sucede entonces podemos obtener una caja mediante la unión del par redundante sin que se afecte la estructura de \mathcal{P} , por lo que en adelante suponemos que \mathcal{P} no contiene pares redundantes de cajas.

Las cajas de S tienen la siguiente clasificación [35]: una caja $B_i \in S$ es una *hoja* si es adyacente solamente a una caja; una caja $B_i \in S$ es una *unión* si existen al menos dos cajas adyacentes a B_i tales que la intersección de esas dos cajas es una arista; una caja $B_i \in S$ es un *conector* si es adyacente a exactamente dos cajas, y la intersección de dichas cajas es vacía. Notemos que, por la suposición de que \mathcal{P} no contiene cajas redundantes, todo conector es adyacente a exactamente dos uniones.

Un *corredor* de \mathcal{P} es una caja contenida en \mathcal{P} que es la unión de al menos dos cajas de S y que no está propiamente contenida en ninguna otra caja de \mathcal{P} , ver

Figura 3.1a.

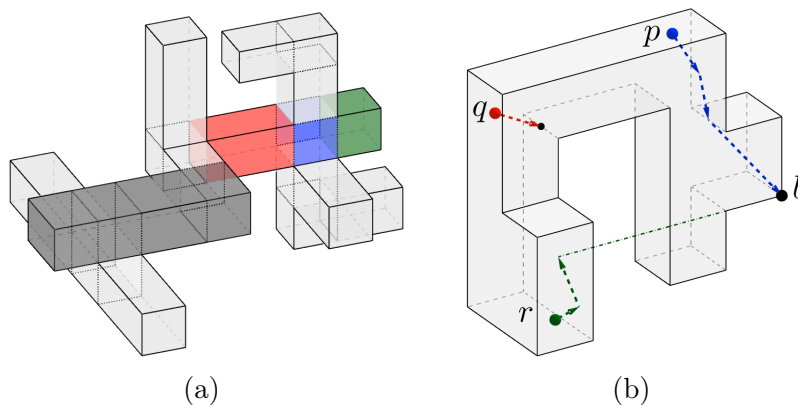


Figura 3.1: a) Ejemplo de un ortoárbol. Las cajas verde, azul y roja son una hoja, una unión y un conector, respectivamente. La caja negra es un corredor. b) Una baliza colocada en el punto b cubre al punto p , pero no cubre a r ni a q . El punto r se queda atorado en una cara. Para el punto q la trayectoria se vuelve indeterminada al tocar una arista, ya que la trayectoria que disminuye localmente la distancia hacia b deja de ser única.

Las siguientes definiciones fueron dadas por Biro en [15] para polígonos, pero aquí las extendemos a poliedros.

Definición 3.3.1. Una *baliza* (del inglés *beacon*) b es colocada en un punto de un polédroo \mathcal{P} y puede ser activada para ejercer una atracción sobre los objetos (representados por puntos) en \mathcal{P} . Cuando b es activada, los objetos en \mathcal{P} se mueven de forma que se minimice localmente su distancia euclidiana hacia b , con la restricción de permanecer en el interior de o la frontera de \mathcal{P} .

Definición 3.3.2. Una baliza *cubre* a un punto p si, bajo la atracción de b , un objeto colocado en p se mueve de modo tal que su distancia Euclidiana con respecto a b disminuya hasta ser cero. De manera equivalente, decimos que p es *cubierto* por b .

3.4. Resultados auxiliares

A continuación se demuestran un lema y un teorema que utilizaremos para demostrar los resultados principales de este capítulo. El lema garantiza que cualquier caja de S contiene al menos cuatro vértices de \mathcal{P} .

Lema 3.4.1. Sea \mathcal{P} un ortoárbol que es la unión de un conjunto de cajas $S = \{B_1, \dots, B_n\}$, y sea B_i una caja de S . Entonces, al menos cuatro vértices de B_i son vértices de \mathcal{P} .

Demostración. Si B_i es una hoja, entonces tiene cuatro vértices que no comparte con ninguna otra caja y por lo tanto son vértices de \mathcal{P} , ver la caja verde en la Figura 3.1a.

Si B_i es un conector, entonces es adyacente a dos uniones, cada una de las cuales es adyacente a otra caja que interseca a B_i en una arista de \mathcal{P} . Los cuatro extremos de esas aristas son vértices de \mathcal{P} , ver la caja roja en la Figura 3.1a.

Si B_i es una unión (ver la caja azul en la Figura 3.1a), entonces es adyacente al menos a dos cajas B_j y B_k tales que $B_j \cap B_k$ es una arista de \mathcal{P} y sus dos extremos son vértices de \mathcal{P} . Si no hay alguna otra caja adyacente a B_i , entonces los dos vértices de B_i que no están contenidos en $B_j \cup B_k$ son vértices de \mathcal{P} . Además, si B_i es adyacente a tres o más cajas entonces incide en al menos cuatro vértices de \mathcal{P} \square

El siguiente teorema nos permite encontrar al menos ocho vértices de \mathcal{P} en cada elemento de un conjunto de corredores de cuya unión se obtiene \mathcal{P} .

Teorema 3.4.2. Cualquier ortoárbol \mathcal{P} contiene un conjunto de corredores $\mathcal{C} = \{C_1, \dots, C_m\}$ tal que $\mathcal{P} = C_1 \cup \dots \cup C_m$ y que cada C_i , $1 < i \leq m$, contiene al menos una cara de una caja de $\mathcal{P}_{i-1} = C_1 \cup \dots \cup C_{i-1}$. Además, para cada elemento $C_i \in \mathcal{C}$ podemos elegir un conjunto V_i de ocho vértices de \mathcal{P} contenidos en C_i tal que:

1. Todos los elementos de V_i contenidos en \mathcal{P}_{i-1} inciden en una misma caja $B \in \mathcal{P}_{i-1}$.
2. Al menos cuatro vértices de V_i no están contenidos en \mathcal{P}_{i-1} , $i \geq 2$.

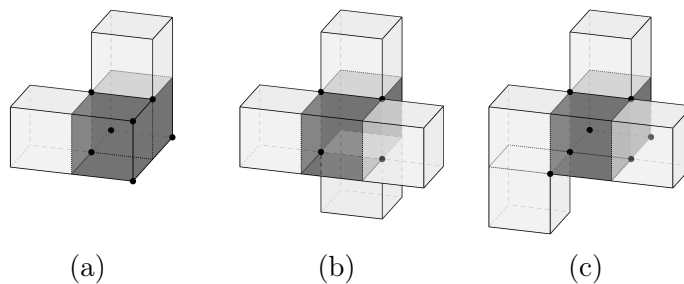


Figura 3.2: Los corredores mostrados en un color más oscuro contienen siete (a), cuatro (b) y siete (c) vértices de \mathcal{P} respectivamente.

Antes de probar este teorema vale la pena señalar que no todos los corredores de un ortoárbol \mathcal{P} contienen ocho vértices de \mathcal{P} , como se muestra en la Figura 3.2. Por lo anterior, la elección del conjunto de corredores del Teorema 3.4.2 debe ser hecha de manera cuidadosa. Observemos también que los corredores que elegimos en la demostración de este teorema pueden intersectarse en su interior, es decir, pueden compartir cajas de \mathcal{S} . Por último, suponemos que \mathcal{P} contiene al menos dos cajas ya que de lo contrario es solamente un prisma rectangular que cumple trivialmente con la propiedad a demostrar.

Demostración. Elegimos los corredores de $\mathcal{C} = \{C_1, \dots, C_m\}$ de forma recursiva como sigue. Sea C_1 un corredor de \mathcal{P} que contiene una hoja, digamos $B_1 \in \mathcal{P}$. Como suponemos que \mathcal{P} contiene al menos dos cajas, existe al menos una caja más en C_1 a la cual llamamos B_2 . Así, B_1 contiene al menos cuatro vértices de \mathcal{P} que no son de B_2 por ser una hoja, y B_2 tiene otros cuatro vértices de \mathcal{P} por el Lema 3.4.1. Por lo tanto, C_1 tiene al menos ocho vértices de \mathcal{P} .

Supongamos que hemos elegido el conjunto de corredores C_1, \dots, C_{k-1} tales que $\mathcal{P}_{k-1} = C_1 \cup \dots \cup C_{k-1}$ es un ortoárbol y que cada corredor del conjunto cumple con la propiedades deseadas.

Si $\mathcal{P}' = \mathcal{P} \setminus \mathcal{P}_{k-1}$ es no vacío, tomamos una caja B' en \mathcal{P}' tal que $B' \cap \mathcal{P}_{k-1}$ es una cara de B' . Sea B la caja de \mathcal{P}_{k-1} que tiene una cara en común con B' , ver Figura 3.3.

Supongamos primero que B' es una hoja de \mathcal{P} . Sea C_k el corredor de \mathcal{P} que contiene a B y a B' . Como B' es una hoja, contiene cuatro vértices de \mathcal{P} que no comparte con ninguna otra caja. Por el Lema 3.4.1, B contiene otros cuatro vértices de \mathcal{P} , los cuales pertenecen a \mathcal{P}_{k-1} . Por lo tanto, C_k contiene al menos ocho vértices, cuatro de los cuales no están en \mathcal{P}_{k-1} .

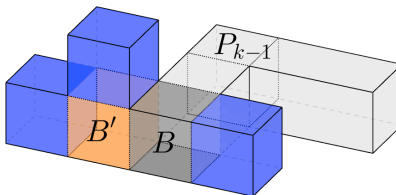


Figura 3.3: Se muestra \mathcal{P}_{k-1} en gris. En amarillo, una caja B' que comparte una cara con una caja B en \mathcal{P}_{k-1} . Las cajas azules no están en \mathcal{P}_{k-1} y podrían estar contenidas en un corredor de \mathcal{P} que contenga a B' .

Supongamos entonces que B' no es una hoja. Elegimos al corredor C_k de \mathcal{P} de la siguiente manera: Si existe un corredor C' de \mathcal{P}' que contenga al menos tres cajas, incluyendo a B' , entonces seleccionamos a C_k como el corredor de \mathcal{P} que contiene a C' . Observemos que, como C' contiene al menos tres cajas, al menos dos de esas cajas con disjuntas, así, por el Lema 3.4.1, C' contiene al menos ocho vértices de \mathcal{P} . Si C' contiene al menos una caja que no interseque a P_{k-1} , entonces cuatro de sus vértices no son vértices de \mathcal{P}_{k-1} . Podría suceder que C' contenga exactamente tres cajas y que las tres intersequen a B . En ese caso, C' tiene a lo más cuatro vértices en común con \mathcal{P}_{k-1} , que son los contenidos en la cara que comparten B y B' . Por lo tanto, podemos elegir ocho vértices de \mathcal{P} en C_k , cuatro de los cuales no están en \mathcal{P}_{k-1} .

Supongamos entonces que cualquier corredor de \mathcal{P}' que contiene a B' tiene a lo más dos cajas. Sea $C' = B' \cup B''$ uno de esos corredores de \mathcal{P}' . Sea C_k el corredor de \mathcal{P} que contiene a C' . Si B pertenece a C_k , entonces C_k contiene al menos tres cajas, una de las cuales, B'' , no interseca a \mathcal{P}_{k-1} . Así, C_k contiene al menos cuatro vértices de \mathcal{P} que no pertenecen a \mathcal{P}_{k-1} (y son vértices de B'').

Supongamos entonces que ningún corredor C' de \mathcal{P}' que contenga a B' está contenido en un corredor de \mathcal{P} que contenga a B .

Sea $C_k = B' \cup B''$ cualquier corredor de \mathcal{P}' que contenga a B' . Sea C_i un corredor de $\{C_1, \dots, C_{k-1}\}$ que contiene a B . Existen dos casos posibles para el número de cajas de C_i con las que B' tiene intersección no vacía: B' interseca exactamente dos cajas de C_i o B' interseca a tres cajas de C_i , ver Figura 3.4.

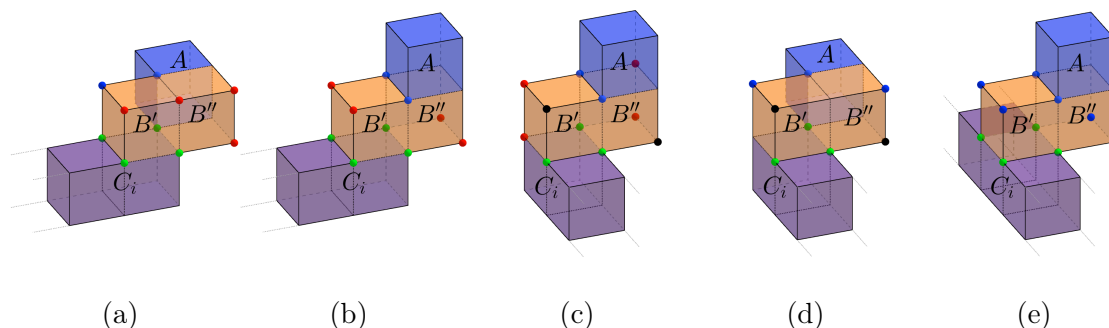


Figura 3.4: a) y b) muestran los casos en los que exactamente una caja en C interseca a $B'' \in C_k$. c) y d) muestran los casos en los que exactamente dos cajas de C intersecan a B'' . e) muestra el caso en el que tres cajas de C intersecan a B'' .

1. Supongamos primero que B' interseca dos cajas de C_i . En este caso, B debe ser un extremo de C_i , como se muestra en la Figura 3.4 de (a) a (d).

Notemos que B'' y C_i se intersecan en una arista de B , ver figura 3.4. Si B'' es una hoja, entonces C_k contiene al menos ocho vértices de \mathcal{P} , cuatro de los cuales pertenecen a B'' y no están en \mathcal{P}_{k-1} , y terminamos.

Supongamos entonces que B'' no es una hoja de \mathcal{P} . Entonces B'' es una unión que en $T_{\mathcal{P}}$ es adyacente a al menos una caja A que no está contenida en \mathcal{P}_{k-1} . Supongamos sin pérdida de generalidad que B' está arriba de B , y que B'' está a la derecha de B' , ver Figura 3.4. Notemos que ninguna caja B'' de \mathcal{P} comparte la cara superior de B' (ya que en tal caso existiría un corredor que contiene a B , B' y B''). Existen dos subcasos posibles:

- 1a. Exactamente una caja contenida en C_i interseca a B'' . En este caso existen cuatro vértices de \mathcal{P} en $C_k \cap C_i$, aquellos de color verde en las figuras 3.4a y 3.4b. Existen dos casos posibles para las cajas adyacentes a B'' en $T_{\mathcal{P}}$.
 - No existe una caja arriba de B'' . Supongamos sin pérdida de generalidad que existe una caja A que comparte la cara trasera de B'' , ver figura 3.4a. Como no existe una caja de \mathcal{P} que comparta la cara superior de B' , los dos vértices que se muestran en azul en la figura 3.4a son vértices de \mathcal{P} . Existe a lo más una caja de \mathcal{P} que comparte una de las caras frontales de las cajas de C_k , por lo que es adyacente a B' o a B'' . En cualquier caso, al menos dos de los vértices rojos mostrados en la figura 3.4a son vértices de \mathcal{P} .
 - Existe una caja A arriba de B'' . En este caso existen dos vértices de \mathcal{P} en $B' \cap A$, los cuales se muestran en azul en la figura 3.4b. Notemos que existen a lo más dos cajas de \mathcal{P} que comparten una cara de C_k , una comparte una cara frontal de C_k y la otra una cara trasera de C_k . En cualquier caso, al menos dos de los vértices rojos mostrados en la Figura 3.4b son vértices de \mathcal{P} .
- 1b. Exactamente dos cajas contenidas en C_i intersecan a B'' , ver figuras 3.4c y 3.4d. En este caso existen tres vértices de \mathcal{P} en $C_k \cap C_i$, los cuales se muestran en verde en las figuras 3.4c y 3.4d. Dado que no existe una caja de \mathcal{P} que comparta alguna de las caras frontales de C_k ni la cara superior de B' , los dos vértices negros mostrados en las figuras 3.4c y 3.4d son vértices de \mathcal{P} . Como B'' no es una hoja, existe una caja A compartiendo una de las caras de B'' . Existen dos casos posibles:

- Existe una caja A que comparte la cara superior de B'' . En este caso existen dos vértices de \mathcal{P} en $B' \cap A$. Puede existir una caja de \mathcal{P} que comparta una de las dos caras posteriores de C_k . Exista o no dicha caja, al menos uno de los vértices rojos mostrados en la Figura 3.4c es un vértice de \mathcal{P} .
 - No existe una caja de \mathcal{P} que comparta la cara superior de B'' . En este caso existe una caja A que comparte la cara posterior de B'' . Como no existe una caja arriba de B' , los tres vértices azules mostrados en la Figura 3.4d son vértices de \mathcal{P} .
2. Exactamente tres cajas contenidas en C_i intersecan a B' , ver Figura 3.4e. En este caso, existe exactamente una caja A de \mathcal{P} que comparte la cara superior de B'' , tal como se muestra en la figura. En este caso C_k contiene diez vértices de \mathcal{P} , como se muestra en la Figura 3.4e.

Es fácil ver que en cada uno de los casos analizados previamente C_k contiene al menos cuatro vértices de \mathcal{P} que no están en \mathcal{P}_{k-1} , lo que termina la prueba. \square

3.5. Cobertura de ortoárboles con balizas

En esta sección presentamos los resultados principales de este capítulo. Decimos que una baliza b cubre a una caja B si cualquier punto $p \in B$ está cubierto por b .

Lema 3.5.1. *Sea \mathcal{P} un ortoárbol y sea C un corredor de \mathcal{P} . Entonces, una baliza b colocada en cualquier punto de C cubre a cualquier caja B cuya intersección con la frontera de C sea no vacía.*

Demostración. Sin pérdida de generalidad supongamos que C está orientado como en la Figura 3.5, que se coloca una baliza b en cualquier parte de C , y que una caja B de \mathcal{P} comparte la cara superior de alguna caja de C . Sea p cualquier punto en B . Cuando b atrae a p , p se mueve en línea recta hasta alcanzar a b o tocar la cara frontal de B , tras lo cual p se deslizará hacia abajo hasta alcanzar C para luego moverse en línea recta hacia b . Por lo tanto, B está cubierto por b .

Consideremos ahora una caja B' que interseca a C en una arista. Podemos suponer sin pérdida de generalidad que B' es adyacente a B , ver Figura 3.5. Sea Q el plano que contiene a $B \cap B'$. Notemos que Q separa el espacio en dos regiones, un semiespacio que contiene a B y b , y un semiespacio que contiene a B' . Además, las caras de B' que tienen un vértice de C son parte de la frontera de \mathcal{P} , con excepción

de la cara $B \cap B'$. Así, cuando b atrae a algún punto $q \in B'$, q se mueve en línea recta hasta b o hasta tocar una cara de B' o de B . En caso de que q toque una cara de B' , entonces se deslizará hasta llegar a B y así, por el caso anterior, q está cubierto por b .

Por último, consideremos una caja B'' cuya intersección con C sea un vértice de B'' . Entonces B'' debe ser adyacente a una caja cuya intersección con C sea una arista de dicha caja, por lo que podemos suponer sin pérdida de generalidad que esa caja es B' . Notemos que las caras de B'' que comparten un vértice con C y no son $B' \cap B''$ deben estar contenidas en la frontera de \mathcal{P} . Notemos también que el plano Q' que contiene a $B' \cap B''$ separa al semiespacio que contiene a b y B' del semiespacio que contiene a B'' . Por lo tanto, cuando un punto $r \in B''$ es atraído hacia b , r se mueve en línea recta hasta alcanzar a B' , o se desliza sobre una cara de B'' que comparte un vértice con C hasta eventualmente alcanzar B' . Por el caso anterior, r está cubierto por b . \square

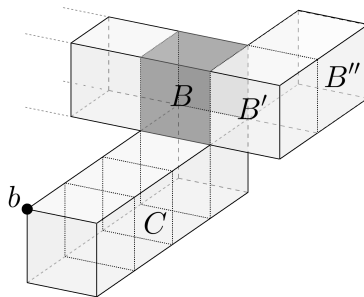


Figura 3.5: Una baliza b colocada en cualquier punto del corredor C cubre a cualquier caja cuya intersección con C sea no vacía, como las cajas B , B' y B'' .

A continuación, demostramos que el conjunto de vértices de un ortoárbol \mathcal{P} se puede colorear con doce colores distintos de forma tal que todo conjunto que contenga los elementos de cualquiera de las clases cromáticas resultantes de esta coloración cubre a \mathcal{P} .

Lema 3.5.2. *Sea \mathcal{P} un ortoárbol para el que se ha seleccionado un conjunto de corredores $\{C_1, C_2, \dots, C_m\}$ como en el Teorema 3.4.2.*

Entonces, los vértices de \mathcal{P} pueden ser coloreados con doce colores distintos de modo tal que cada una de las siguientes condiciones se cumple:

1. Los vértices de \mathcal{P} que inciden en la misma caja B y han sido coloreados tienen distinto color.
2. Los vértices de \mathcal{P} en cada corredor C_i se han coloreado de forma tal que existen al menos ocho vértices de \mathcal{P} en C_i con colores distintos.
3. Por cada corredor $C_j \in \{C_2, \dots, C_m\}$ existe otro corredor C_i , $i < j$ que interseca a C_j de forma tal que $C_i \cup C_j$ contiene doce vértices de \mathcal{P} coloreados con colores distintos.

Demostración. Comenzamos con la observación de que los vértices del primer corredor C_1 se pueden colorear de manera que las condiciones 1 y 2 se cumplan, ya que cada caja de S incide en a lo más ocho vértices de \mathcal{P} .

Supongamos que los vértices de \mathcal{P} en \mathcal{P}_{j-1} , $j \geq 2$, han sido coloreados de tal forma que se satisfacen las condiciones 1, 2 y 3.

Según la prueba del Teorema 3.4.2, al momento de seleccionar el corredor C_j elegimos una caja B' que no estaba contenida en \mathcal{P}_{j-1} , pero cuya intersección con \mathcal{P}_{j-1} es una cara de una caja $B \in \mathcal{P}_{j-1}$.

Sea C_i un corredor en $\{C_1, \dots, C_{j-1}\}$ que contiene a B . Por la hipótesis de inducción, C_i contiene un conjunto de ocho vértices coloreados con colores distintos. Elegimos un conjunto V' que cumpla lo anterior y que contenga a los vértices de \mathcal{P} en B . Supongamos sin pérdida de generalidad que los elementos de V' están coloreados con los colores del 1 al 8. Por el Teorema 3.4.2, C_j contiene al menos cuatro vértices de \mathcal{P} que no están en \mathcal{P}_{j-1} . Ahora elegimos un conjunto V'' de cuatro vértices de \mathcal{P} en C_j que contenga el mayor número posible de vértices sin colorear incidentes en B' ; coloreamos estos vértices con los colores del 9 al 12. De esta manera, la coloración de los vértices de \mathcal{P} que inciden en $B' \cup V''$ puede ser extendida a una coloración de los vértices de \mathcal{P} en C_j que cumpla con las condiciones 1, 2 y 3. \square

Con los lemas anteriores podemos probar el resultado principal de este capítulo. La cota inferior del siguiente teorema fue presentada en la tesis de maestría del autor [74], sin embargo, se incluye aquí con el fin mostrar un panorama completo del problema en cuestión.

Teorema 3.5.3. *Sea \mathcal{P} un ortoárbol con n vértices. Entonces, $\lfloor n/12 \rfloor$ balizas colocadas en vértices de \mathcal{P} son siempre suficientes para cubrir a \mathcal{P} . Además, $\lfloor n/12 \rfloor$ balizas son a veces necesarias para cubrir a \mathcal{P} .*

Demostración. Primero demostramos la cota superior. Sea B' una caja de \mathcal{P} y sea C_j el corredor con menor j que contiene a B' . Existen dos casos a considerar, dependiendo de si B' interseca a \mathcal{P}_{j-1} o no.

Si B' interseca a \mathcal{P}_{j-1} , entonces existe una caja $B \in \mathcal{P}_{j-1}$ que comparte una cara, una arista o un vértice con B' . Al colorear los vértices de C_j elegimos un corredor C_i , con $i < j$, tal que $C_i \cup C_j$ contiene doce vértices de \mathcal{P} coloreados con colores distintos. Claramente, $B \in C_i$, y así, por el Lema 3.5.1 se cumple que cualquier vértice de C_i cubre a B' , por lo que B' está cubierto por al menos un vértice de cada color.

Supongamos que B' no interseca a \mathcal{P}_{j-1} . Por la suposición de que \mathcal{P} no contiene cajas redundantes se cumple que B' interseca al menos una caja B'' que no pertenece a C_j (B' puede intersecar a B'' en una cara, una arista o un vértice). Así, B'' pertenece a un corredor C_r , $r > j$. Elegimos a la caja B'' de tal modo que r sea tan pequeño como sea posible. Cuando coloreamos los vértices de C_r elegimos un corredor C' en $\{C_1, \dots, C_{r-1}\}$ tal que los vértices de ese corredor unidos con los vértices de C_r tengan doce vértices coloreados con colores distintos. Dado que elegimos a C_r con r tan pequeño como sea posible, C' tiene que ser C_j . Así, por el Lema 3.5.1 se cumple que B' está cubierto por todos los vértices de $C_j \cup C_r$, y por lo tanto, B' está cubierto por un vértice de cada color. Dado que existe al menos una clase cromática con a lo más $\lfloor n/12 \rfloor$ vértices de \mathcal{P} , se cumple que $\lfloor n/12 \rfloor$ balizas colocadas en los vértices \mathcal{P} son suficientes para cubrir a \mathcal{P} .

Ahora demostramos la cota inferior. Bae et al. [9] probaron que existe un polígono ortogonal con forma de espiral \mathcal{R} de m vértices necesita $\lfloor m/6 \rfloor$ balizas para ser cubierto. Esto implica que el ortoárbol espiral \mathcal{P} cuyas caras inferior y superior son copias de \mathcal{R} y tiene $n = 2m$ vértices necesita $\lfloor n/12 \rfloor$ balizas para ser cubierto, dado que \mathcal{P} necesita al menos tantas balizas como \mathcal{R} . Un ejemplo del polígono de Bae et al. y del ortoárbol cuyas caras inferior y superior son dicho polígono se muestra en la Figura 3.6b. \square

3.6. Complejidad del problema de la galería de arte en ortoárboles

Es sabido que el problema de encontrar un conjunto mínimo de balizas que cubra a un poliedro general es un problema *NP*-difícil [28]. Por otro lado, no se ha probado a la fecha que este problema sea *NP*-difícil en ortoárboles. Sin embargo, podemos utilizar resultados conocidos en \mathbb{R}^2 para probar que el problema de encontrar un conjunto

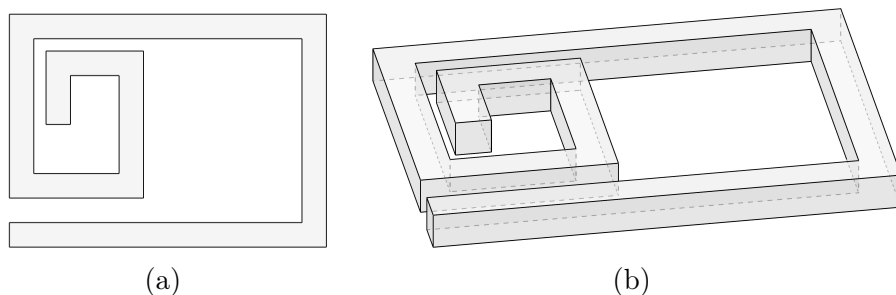


Figura 3.6: a) Un polígono ortogonal \mathcal{P} que necesita $\lfloor m/6 \rfloor$ balizas para ser cubierto. b) Un ortoárbol cuyas caras superior e inferior son copias de \mathcal{P} , el cual necesita $\lfloor n/12 \rfloor$ balizas para ser cubierto.

mínimo de guardias es NP -difícil en ortoárboles, por lo que podríamos esperar que este también fuera el caso para encontrar un conjunto mínimo de balizas.

Teorema 3.6.1. *El problema de encontrar un conjunto mínimo de vértices que vigile un ortoárbol es NP -difícil. Este problema es incluso APX -difícil.*

Demostración. Se dice que un polígono ortogonal \mathcal{Q} es *delgado* si la gráfica dual de la partición obtenida al extender las aristas de \mathcal{Q} hacia el interior de \mathcal{Q} hasta que toquen una arista es un árbol [98]. Se muestra un ejemplo en la Figura 3.7. Tomás [98] probó que el problema de encontrar un conjunto mínimo que vigile un polígono delgado es NP -difícil y APX -difícil. Lo anterior implica que la complejidad de vigilar un ortoárbol con un conjunto guardia mínimo es también NP -difícil y APX -difícil. Para demostrar esto, tomemos cualquier polígono ortogonal delgado \mathcal{P} en un plano ortogonal al eje z , y tomemos una copia \mathcal{P}' de \mathcal{P} trasladada hacia arriba. Notemos que el poliedro ortogonal \mathcal{P} tal que su cara inferior es \mathcal{P} y su cara superior es \mathcal{P}' es un ortoárbol. Así, \mathcal{P} necesita al menos tantos guardias como \mathcal{P} para ser vigilado, ya que para que \mathcal{P} esté completamente vigilado es necesario que \mathcal{P} esté a su vez vigilado. Además, cualquier conjunto de puntos que vigile a \mathcal{P} es suficiente para vigilar a \mathcal{P} y al trasladar cualquier subconjunto del conjunto guardia verticalmente \mathcal{P} continúa siendo vigilado. Por lo tanto, vigilar ortoárboles es al menos tan difícil como vigilar polígonos ortogonales delgados. \square

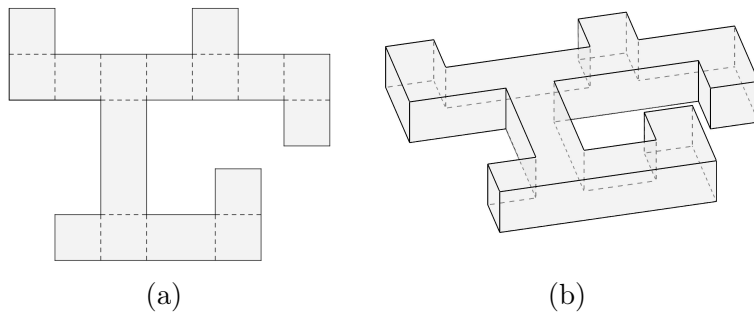


Figura 3.7: a) Un polígono ortogonal delgado P . b) Un ortoárbol cuyas caras superior e inferior son copias de P .

Capítulo 4

Incrustaciones de árboles en mallas rectangulares

4.1. Introducción

Consideremos un árbol T cuyos vértices tienen grado máximo cuatro. Es fácil darnos cuenta de que T puede ser dibujado en el plano de modo tal que sus vértices sean representados como puntos con coordenadas enteras, y que sus aristas sean representadas mediante segmentos verticales u horizontales con interiores disjuntos. Llamamos a este tipo de representación un *encaje de T en la malla*.

Existen varias propiedades de los encajes de gráficas en la malla que se han considerado en aplicaciones como la *integración a muy gran escala* o *VLSI* (del inglés, *very large-scale integration*), que consiste en crear circuitos integrados compuestos por un gran número de componentes, de modo tal que se satisfagan ciertos requerimientos eléctricos y de diseño. Una propiedad que se ha considerado en diversas publicaciones es el área del encaje, la cual suele estar dada por el área del rectángulo alineado a los ejes más pequeño que contiene al encaje en cuestión. La propiedad en la que nos enfocamos en este capítulo es el número de *dobleces*, es decir, el número de veces que una arista representada por un segmento horizontal y otra representada por un segmento vertical son adyacentes.

Notemos que, a diferencia de lo que sucede con los árboles, la condición de que sus vértices tengan grado máximo cuatro no es suficiente para que una gráfica *aplanable* tenga un encaje en el que sus aristas sean representadas por segmentos horizontales o

verticales. Por esta razón, diversos autores han estudiado variantes de este problema en las que se permite que las artistas sean representadas por trayectorias poligonales formadas de segmentos horizontales y verticales. En este contexto se han desarrollado algoritmos para encontrar encajes en los que se minimice la cantidad máxima de dobleces de cada arista. Por otra parte, se han estudiado encajes de gráficas en las que se minimice la cantidad total de dobleces en el encaje, ya sea que se permitan o no dobleces en las trayectorias que representan a las aristas.

En un encaje de un árbol T en la malla cada trayectoria P de T está representada por una sucesión de segmentos horizontales y verticales, por lo que podemos contar el número de dobleces de cada trayectoria. Notemos que pueden existir trayectorias con cero dobleces, que son aquellas formadas únicamente por segmentos horizontales o verticales. Además, el número de dobleces de una misma trayectoria puede variar dependiendo del encaje. Un ejemplo de esto se muestra en la Figura 4.1, la cual contiene dos encajes diferentes de un mismo árbol. En el encaje que se muestra en la Figura 4.1a la trayectoria que une a los vértices i y r tiene cuatro dobleces, mientras que en la Figura 4.1b esa misma trayectoria tiene tres dobleces. Luca, Oliveira y Szwarcfiter [37] propusieron el problema de encontrar un encaje de un árbol en la malla en el que se minimice el máximo número de dobleces de las trayectorias de T . En ese artículo presentaron un algoritmo de tiempo $O(n^2)$ para encontrar dichos encajes.

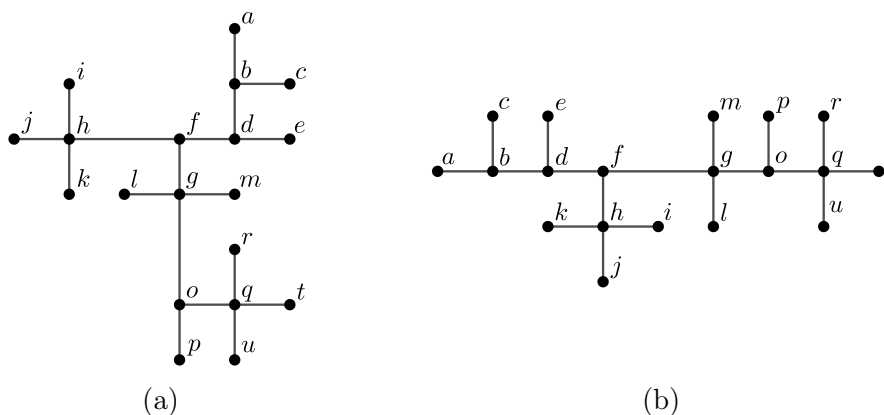


Figura 4.1: Dos encajes distintos de un árbol T en la malla entera.

En este capítulo presentamos un algoritmo de tiempo $O(n)$ que, dado un árbol T de grado máximo cuatro, encuentra un encaje que minimiza el máximo número de dobleces entre todas las trayectorias de T . Este algoritmo puede generalizarse para encontrar encajes de árboles de grado $2d$ en \mathbb{R}^d .

4.2. Definiciones

Una *gráfica* G es una tupla $(V(G), E(G))$, donde $V(G)$ es un conjunto de elementos llamados *vértices* y $E(G)$ es un conjunto de parejas de vértices llamadas *aristas*. Nos referiremos a $V(G)$ y $E(G)$ simplemente como V y E cuando no exista ambigüedad.

Dada una arista $e = (u, v)$ decimos que e *une* a u con v y que e *incide* en u y en v . El número de aristas que inciden en un vértice v se conoce como el *grado* de v .

Una gráfica F es llamada una *subgráfica* de G si $V(F) \subseteq V(G)$ y $E(F) \subseteq E(G)$. Decimos que G *contiene* a F , o que F está *contenida* en G , lo cual se escribe $G \supseteq F$ o $F \subseteq G$.

Un *camino* es una secuencia lineal de vértices en una gráfica, tal que existe una arista entre cada vértice de la secuencia y el vértice siguiente. Una *trayectoria* es un camino que no repite vértices. Decimos que una trayectoria une a cualesquiera dos de los vértices que la componen. Un *ciclo* en tres o más vértices es como una trayectoria salvo que el primer y el último vértice son el mismo. La *longitud* de una trayectoria o un ciclo es igual al número de aristas que los componen. La *distancia* $d(p, q)$ entre dos vértices p y q es la longitud de la trayectoria más corta que los une.

Una gráfica es *conexa* si para cualesquiera dos de sus vértices existe una trayectoria que los une; de otro modo, la gráfica es llamada no conexa. Una gráfica conexa que no contiene ciclos es llamada un *árbol*.

Una gráfica *aplanable* es aquella que puede ser representada en el plano de modo tal que sus aristas (representadas por curvas) sólo se intersequen en sus extremos comunes (representados por puntos), a tal representación se le conoce como *encaje plano*. A una gráfica aplanable junto con su encaje plano se le conoce como gráfica *plana*. Notemos que todo árbol es aplanable.

Sea G una gráfica aplanable. Un *encaje en la malla* de G (también llamado *encaje ortogonal* de G) es un encaje plano de G tal que sus vértices son representados como puntos con coordenadas enteras y sus aristas son segmentos de línea verticales u horizontales con interiores disjuntos. Se muestra un ejemplo de esto de la Figura 4.1. Observemos que es condición necesaria que cada uno de los vértices de G tenga grado a lo más cuatro para que G tenga un encaje en la malla, ya que de otro modo no existirán orientaciones suficientes para representar todas las aristas incidentes en cualquier vértice de grado mayor.

Sea $T = (V, E)$ un árbol y sea \mathcal{M} un encaje de G en la malla. Dada una trayectoria

P de T decimos que el número de dobleces de P en un encaje \mathcal{M} es el número de pares de aristas consecutivas de P en los que una es vertical y la otra es horizontal.

4.3. Trabajo relacionado

El problema de encontrar encajes de gráficas en las que sus vértices sean representados por puntos en el plano y sus aristas como segmentos de línea recta, conocidos usualmente como *encajes rectilíneos*, han sido estudiadas desde hace más de siete décadas. En 1948, István Fáry [51] demostró que toda gráfica aplanable tiene un encaje rectilíneo en el plano. Desde entonces se han estudiado varios problemas relacionados. Una variante natural es pedir que los puntos que representan a los vértices tengan coordenadas enteras. En este sentido, varios autores han estudiado el problema de encontrar encajes con la mínima área posible. Crescenzi, Di Battista y Piperno [32] demostraron que todo árbol binario completo tiene un encaje rectilíneo en una malla de tamaño lineal, el cual puede ser encontrado en tiempo lineal. De Fraysseix, Pach y Pollack [36] demostraron que toda gráfica aplanable con $n \geq 3$ vértices tiene un encaje rectilíneo en la malla entera de tamaño $(2n - 4) * (n - 2)$, y presentaron un algoritmo de tiempo $O(n \log n)$ para encontrar dichos encajes. Recientemente, Chan [19] demostró que los árboles binarios pueden ser encajados en mallas enteras de área $n2^{\mathcal{O}(\log^* n)}$, y Covella, Frati y Patrignani [30] probaron que los árboles ternarios completos pueden ser encajados en mallas enteras de área $\mathcal{O}(n^{1.149})$.

En cuanto a los encajes ortogonales en la malla, Garg y Tamassia [56] probaron que dada una gráfica plana con grado máximo cuatro, decidir si admite un encaje ortogonal es un problema NP -difícil. Este no es el caso para las gráficas planas-exterioras y las gráficas en serie-paralelas. Se puede encontrar un encaje ortogonal o determinar que este no existe en tiempo lineal, como lo demostraron Frati [55] para el caso de las gráficas planas-exterioras, y Didimo et al. [41] para el caso de las gráficas en serie-paralelas.

Consideremos ahora el caso en el que se permite que las aristas sean trayectorias ajenas compuestas de segmentos horizontales y verticales. Cualquier gráfica aplanable de grado máximo cuatro admite un encaje de este tipo en el que cada arista tiene a lo más cuatro dobleces, véanse [90] y [101]. Un algoritmo de tiempo lineal para encontrar dichos encajes fue presentado por Tamassia y Tollis [93]. Después, Liu et al. [72] presentaron un algoritmo de tiempo lineal para encontrar encajes de estas gráficas en las que cada arista tiene a lo más dos dobleces, a excepción del octaedro, ya que cualquier encaje del octaedro tiene una arista con al menos tres dobleces. Para gráficas cúbicas (gráficas en las que todos sus vértices tienen grado exactamente 3),

Yanpei et al. [106] probaron que cualquier gráfica cúbica, a excepción del octaedro, admite un encaje en el que cada arista tiene exactamente un doblez, y presentaron un algoritmo de tiempo lineal para encontrar dicho encaje. Las gráficas para las cuales existe un encaje ortogonal con a lo más un doblez por arista fueron caracterizadas por Morgana et al. [78], quienes también presentaron un algoritmo de tiempo $O(n^2)$ para encontrar dicho encaje.

Tamassia [92] presentó un algoritmo de tiempo $O(n^2 \log n)$ que, dada una gráfica plana G de grado máximo cuatro, encuentra un encaje ortogonal isomorfo a G que preserva sus regiones y que minimiza el número de dobleces de sus aristas. Di Battista et al. [40] presentó algoritmos que minimizan el número total de dobleces en encajes ortogonales. En el caso de las gráficas en serie-paralelas aplanables de grado máximo cuatro presentaron un algoritmo de tiempo $O(n^4)$ y en el caso de las gráficas en serio paralelas cúbicas aplanables presentaron un algoritmo de tiempo $O(n^3)$.

4.4. Resultados

Sea T un árbol de grado máximo cuatro. En adelante suponemos que T no tiene aristas de grado dos, puesto que siempre existe un encaje ortogonal en el que se minimice el número máximo de dobleces de una trayectoria tal que las aristas adyacentes a vértices de grado dos estén alineadas.

Nos referimos como un *modelo* de T al par formado por el árbol T y una colección de *órdenes locales* $\{\ell_v\}_{v \in V}$, en donde ℓ_v indica los ángulos entre aristas adyacentes que inciden en un vértice v . De esta forma se almacena en ℓ_v la descripción del encaje alrededor de v , sujeto a rotación.

Notemos que cualquier encaje ortogonal de T tiene asociado un modelo de T , que es precisamente el modelo que corresponde a los órdenes en los que las aristas están dibujadas alrededor de cada vértice en dicho encaje; decimos que el encaje de T es una *representación* de dicho modelo. Se muestra un ejemplo de esto en la Figura 4.2. Nuestra estrategia toma en cuenta lo anterior, pues consiste en encontrar un modelo de T para después encontrar un encaje ortogonal de T que sea una representación de ese modelo. Primero mostraremos que, dado el modelo, podemos encontrar el encaje en tiempo $O(n)$.

Lema 4.4.1. *Para todo modelo \mathcal{M} , existe un encaje ortogonal que es representación de \mathcal{M} . Además, dicho encaje se puede encontrar en tiempo $O(n)$ a partir de \mathcal{M} .*

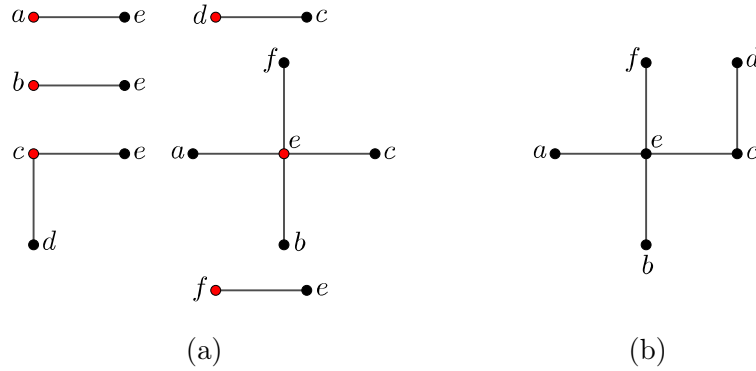


Figura 4.2: (a) Un modelo \mathcal{M} que consiste de seis estrellas, las cuales representan los órdenes locales alrededor de los vértices $\{a, b, c, d, e, f\}$. (b) Una representación de \mathcal{M} . Notemos que algunas de las estrellas en (a) deben ser rotadas para estar propiamente alineadas. Además, las aristas de la representación de un modelo pueden tener longitudes distintas.

Demostración. Elegimos un vértice arbitrario v_0 y partimos V en conjuntos L_i , $i = 1, \dots, t$, donde $L_i = \{v \in V \mid d(v_0, v) = i\}$ contiene todos los vértices a distancia i de v_0 . Después realizamos un proceso que consiste en t pasos. En el paso i , dibujamos todas las aristas que tienen uno de sus extremos en L_{i-1} y el otro extremo en L_i como segmentos de longitud 2^{t-i} de manera que se respeten los órdenes locales de \mathcal{M} correspondientes a las aristas alrededor de los vértices en L_{i-1} . Esto es posible porque a lo más una de las aristas incidentes en cada uno de estos vértices ha sido dibujada hasta este punto. En la Figura 4.3 se ejemplifica un paso de este proceso.

Resta mostrar que cualesquiera dos aristas sin extremos en común tienen intersección vacía. Esto se cumple por la asignación de las longitudes de los segmentos que representan a las aristas, dado que $2^0 + 2^1 + \dots + 2^{k-1} < 2^k$. De este modo, un encaje ortogonal que es representación de \mathcal{M} puede ser encontrado en tiempo $O(n)$. \square

Observemos que el número de dobleces de una trayectoria en un encaje de T depende únicamente del modelo del cual es representación. A continuación, mostramos la manera de encontrar un modelo que minimice el máximo número de dobleces entre las trayectorias de T .

Sea \mathcal{M} un modelo de T . Denotamos con $b(\mathcal{M})$ al máximo número de dobleces entre todas las trayectorias de T en cualquier encaje que sea una representación de \mathcal{M} . Sea $b(T)$ el mínimo valor $b(\mathcal{M})$ entre todos los modelos de T . Nuestro objetivo

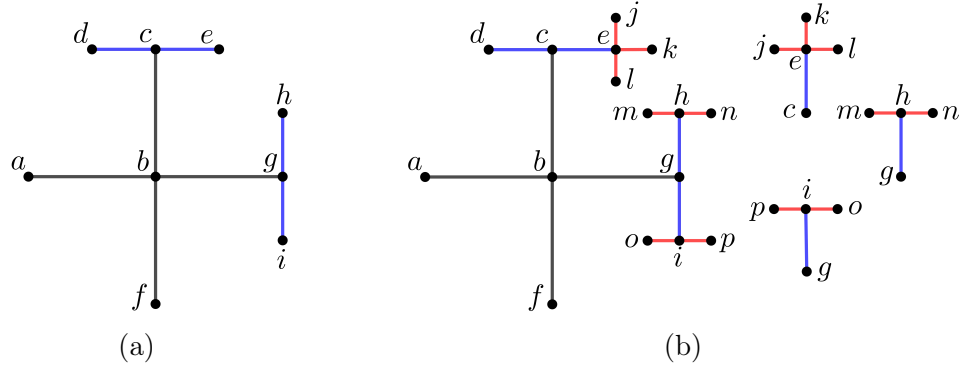


Figura 4.3: Un ejemplo de un paso en la representación de un modelo de T .

es determinar $b(T)$ y encontrar un modelo para el que se alcance este valor.

Dada una arista $e = (u, v)$ definimos como $T[e, u]$ al subárbol resultante de la unión de todas las trayectorias que contienen a e y tienen por extremos a v y a una hoja del árbol. Se muestra un ejemplo en la Figura 4.4. Denotamos con $b_{\mathcal{M}}(e, u)$ al máximo número de dobleces entre todas las trayectorias de $T[e, u]$ que contienen a e en el modelo \mathcal{M} . Asimismo, denotamos con $b(e, u)$ al mínimo valor de $b_{\mathcal{M}}(e, v)$ entre todos los modelos de T . Nos referiremos a $b(e, u)$ como el valor asociado a e cuando se orienta de u a v , y a $b(e, v)$ como el valor asociado a e cuando se orienta de v a u . Decimos que un modelo \mathcal{M} de T es óptimo para $b(e, u)$ si se cumple $b_{\mathcal{M}}(e', u') = b(e', u')$ para todo par (e', u') tal que $T[e', u'] \subseteq T[e, u]$ y que e' esté orientado hacia v .

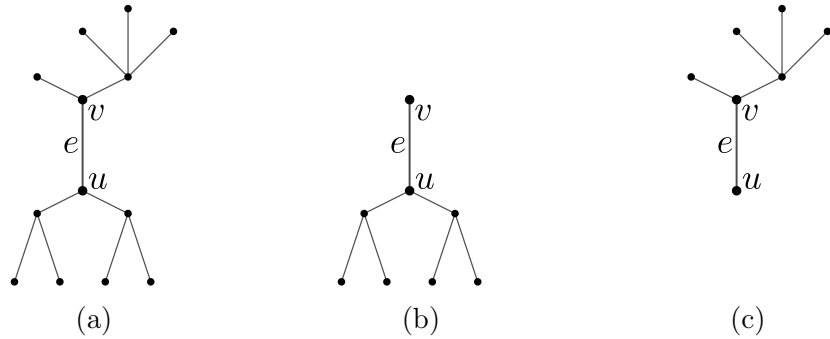


Figura 4.4: (a) Un árbol T y una arista $e = (u, v)$. (b) $T[e, u]$. (c) $T[e, v]$.

Notemos que cada modelo de T induce un modelo de $T[e, v]$, el cual determina el número de dobleces de todas las trayectorias en este subárbol. A continuación

mostramos que se pueden obtener todos los $2n - 2$ valores de la forma $b(e, v)$, correspondientes a las dos orientaciones de cada arista de T , en tiempo lineal. Sea v un vértice de T y sean $e_1 = (u_1, v)$, $e_2 = (u_2, v), \dots, e_d = (u_d, v)$ las aristas incidentes a v , $d \in \{3, 4\}$.

Lema 4.4.2. *Dados los valores $b(e_1, u_1), \dots, b(e_{d-1}, u_{d-1})$ y sus modelos óptimos, \mathcal{M}_1 de $T[e_1, u_1], \dots, \mathcal{M}_{d-1}$ de $T[e_{d-1}, u_{d-1}]$, respectivamente, es posible computar $b(e_d, v)$ y un modelo óptimo de $T[e_d, v]$ en tiempo constante.*

Demostración. Suponemos s.p.d.g. que $b(e_1, u_1) \leq \dots \leq b(e_{d-1}, u_{d-1})$. Para minimizar el número de cruces de las trayectorias de $T[e_d, v]$ que terminan en e_d es necesario que e_d quede alineado con e_{d-1} , lo cual implica que $b(e_d, v) \geq \max\{b(e_{d-2}, u_{d-2}) + 1, b(e_{d-1}, u_{d-1})\}$. Además, $T[e_d, v] \setminus \{(u_d, v)\} = T[e_1, u_1] \cup \dots \cup T[e_{d-1}, u_{d-1}]$ y esos $d - 1$ árboles comparten solamente el vértice v . Por lo tanto, para unir los modelos $\mathcal{M}_1, \dots, \mathcal{M}_{d-1}$ basta con asignar el orden local ℓ_v , y podemos asignar dicho orden de modo tal que (u_{d-1}, v) y (v, u_d) estén alineadas, obteniendo así un modelo óptimo \mathcal{M} de $T[e_d, v]$. Se muestra un ejemplo en la Figura 4.5. \square

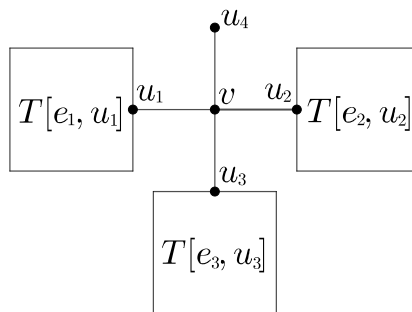


Figura 4.5: Configuración para $d = 4$ en la prueba del Lema 4.4.2. Notemos que u_3 , v y u_4 yacen sobre la misma recta.

A continuación, asignamos una hoja arbitraria v de T orientamos a sus aristas de manera que apunten hacia v .

Teorema 4.4.3. *Todos los $2(n - 1)$ valores de la forma $b(e, v)$, así como el conjunto de los $2(n - 1)$ modelos óptimos de sus subárboles correspondientes pueden ser obtenidos en tiempo $O(n)$.*

Demostración. Obtenemos los $2(n - 1)$ valores en dos etapas, ilustradas en la Figura 4.6.

Para computar los $n - 1$ valores correspondientes a las aristas dirigidas hacia v creamos una pila en la que insertamos todas las aristas dirigidas cuya cola sea una hoja. Mientras exista un elemento $e = (u, w)$ al tope de la pila hacemos lo siguiente: Si u es una hoja asignamos el valor $b(e, u) = 0$. De otro modo, los valores de todas las aristas dirigidas que apuntan hacia u y los modelos óptimos de sus subárboles correspondientes han sido computados en una iteración anterior, y por el Lema 4.4.2 podemos obtener $b(e, u)$ así como el modelo óptimo de $T[e, u]$ en tiempo constante. Agregamos a la pila una arista dirigida $e = (u, w)$ cuando los valores de todas las aristas que apuntan a u han sido obtenidos.

En la segunda etapa orientamos todas las aristas desde v . Sea $e' = (v, w)$ la arista dirigida que contiene a v como cola. Asignamos $b(e', v) = 0$ y realizamos una búsqueda de anchura desde w . Notemos que para cada vértice x visitado en la búsqueda ya han sido obtenidos los valores de todas las aristas que apuntan a x así como los modelos óptimos de sus subárboles correspondientes. Por lo tanto, podemos aplicar el Lema 4.4.2 para obtener el valor de la arista dirigida con cola x en tiempo constante. \square

Utilizaremos el siguiente resultado para probar que, dados los valores computados anteriormente, podemos encontrar un modelo óptimo en tiempo lineal.

Lema 4.4.4. *Existe una arista $e_M = (a, b)$ de T tal que $b(e_M, a) \leq b(e_M, b) \leq b(e_M, a) + 1$. Además, si $b(e_M, a) \neq b(e_M, b)$, entonces se cumple que $b(e, v) < b(e_M, b)$ para cada par (e, v) con $T[e, v] \subset T[e_M, b]$. Dados todos los valores de la forma $b(e, v)$ podemos encontrar dicha arista en tiempo $\mathcal{O}(n)$.*

Demostración. Sea max el máximo de los valores de la forma $b(e, v)$. Sea $f = (u, v)$ una arista dirigida tal que $b(f, v) = max$ y que las aristas dirigidas que apuntan a v tienen valor a lo más $max - 1$. Por la forma de asignar dichos valores la arista f existe, además de que podemos encontrarla en tiempo $\mathcal{O}(n)$ ya que T es un árbol. Si $b(f, u) \geq max - 1$ terminamos, ya que por la elección de f se cumple $b(e, w) < b(f, v)$ se cumple para cualquier par (e, w) con $T[e, w] \subset T[f, v]$, ver Figura 4.7a. Supongamos que $b(f, u) < max - 1$. Consideremos el caso donde el grado de v es 3. Necesariamente, ambas aristas incidentes en v distintas a f tiene en ambas orientaciones el valor $max - 1$, ver Figura 4.7b. Si el grado es cuatro hay dos casos dependiendo del valor que apunta a v desde la cuarta arista. Si el valor es menor a

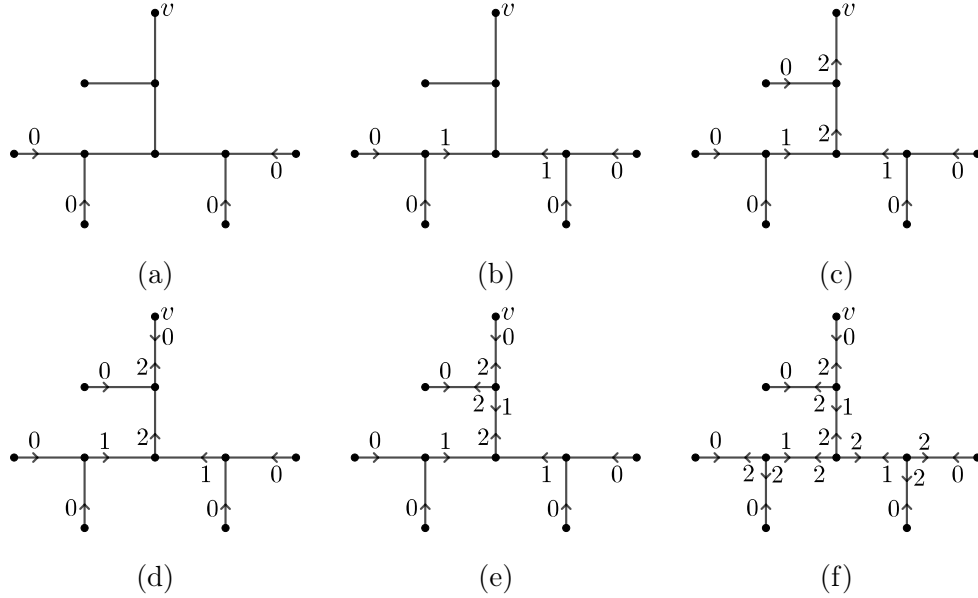


Figura 4.6: Un ejemplo del proceso de la obtención de los valores de la forma $b(e, u)$ para un árbol. En las figuras de (a) a (c) se muestran algunos pasos de la primera etapa y en las figuras de (d) a (f) se muestran algunos pasos de la segunda etapa.

$max - 1$ es similar al caso anterior. En otro caso, tenemos tres aristas con valores max y $max - 1$ que cumplen con la propiedad deseada, ver Figura 4.7c.

□

4.4.1. Algoritmo

Primero obtenemos todos los los valores de la forma $b(e, v)$ y sus modelos correspondientes como en el Teorema 4.4.3. Después, encontramos una arista $e_M = (a, b)$ como la descrita en el Lema 3 y tomamos los modelos óptimos \mathcal{M}_1 de $T[e_M, a]$ y \mathcal{M}_2 de $T[e_M, b]$. Estos modelos puede ser unidos en un modelo \mathcal{M} que sea óptimo con respecto a $b(e_M, a)$ y $b(e_M, b)$. Por último, obtenemos una representación de \mathcal{M} utilizando el Lema 4.4.1.

Sea $B = b(e_M, a) + b(e_M, b)$. Vamos a demostrar que $b(\mathcal{M}) = B = b(T)$, lo que implica que el encaje obtenido minimiza los dobles entre todas las trayectorias de T .

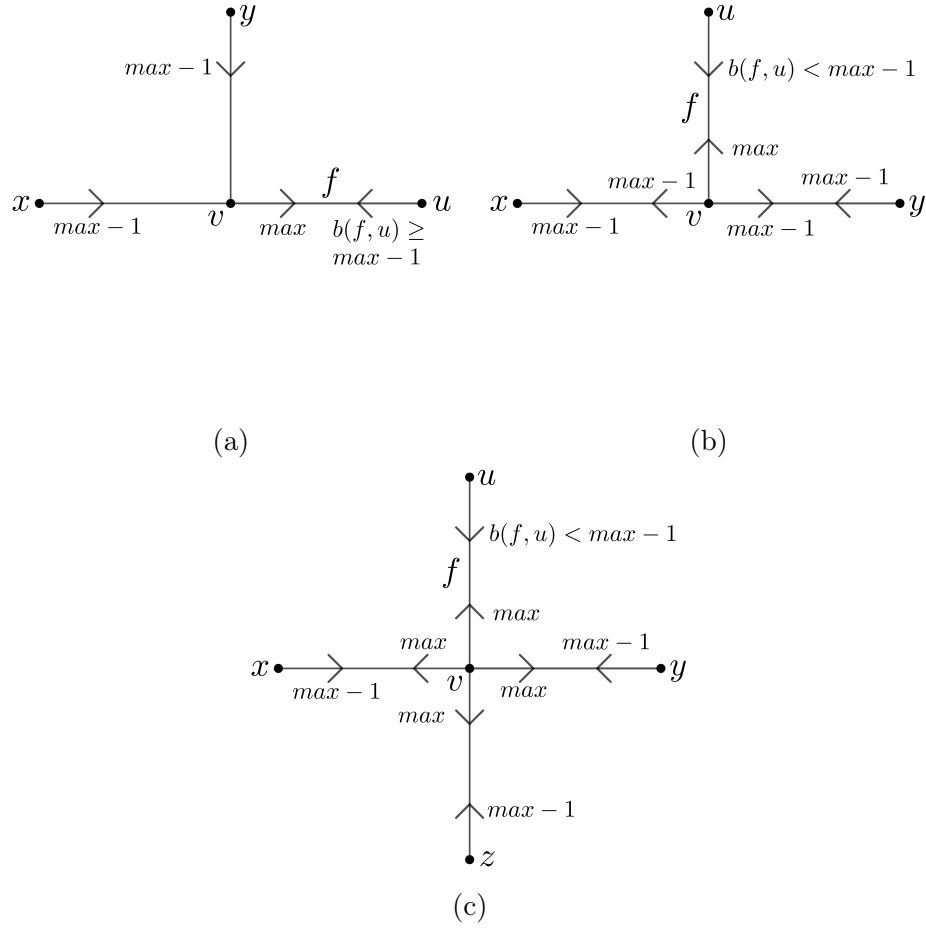


Figura 4.7: Ilustración de la prueba del Lema 4.4.4

Teorema 4.4.5. *El modelo \mathcal{M} obtenido a partir de nuestro algoritmo es tal que $b(\mathcal{M}) = B = b(T)$.*

Demostración. Primero, notemos que existen dos vértices $x \in V[e_M, a]$ y $y \in V[e_M, b]$ tales que $b_{\mathcal{M}}(x, a) \geq b(e_M, a)$ y $b_{\mathcal{M}}(y, b) \geq b(e_M, b)$. Por lo tanto, $b_{\mathcal{M}}(x, y) \geq b(e_M, a) + b(e_M, b) = B$ y $b(T) \geq B$.

Falta demostrar que $b(\mathcal{M}) \leq B$. Para ello, recordemos que por el Lema 4.4.4, $b(e_M, a) \leq b(e_M, b) \leq b(e_M, a) + 1$, y que si $b(e_M, a) \neq b(e_M, b)$, entonces se cumple que $b(e, v) < b(e_M, b)$ siempre que tengamos $T[e, v] \subset T[e_M, b]$. Además, \mathcal{M} es óptimo con respecto a $b(e_M, a)$ y $b(e_M, b)$. Sean $p, q \in V$ dos vértices arbitrarios. Mostraremos que $b_{\mathcal{M}}(p, q) \leq B$, para lo cual tenemos tres casos posibles:

Caso 1: $p \in T[e_M, a]$ y $q \in T[e_M, b]$, ver la Figura 4.8 para un ejemplo. Tenemos que $b_{\mathcal{M}}(p, q) = b_{\mathcal{M}}(p, b) + b_{\mathcal{M}}(q, a)$. Como \mathcal{M} es óptimo con respecto a $b(e_M, a)$ y $b(e_M, b)$, $b_{\mathcal{M}}(p, b) \leq b_{\mathcal{M}}(e_M, a) = b(e_M, a)$ y $b_{\mathcal{M}}(q, a) \leq b_{\mathcal{M}}(e_M, b) = b(e_M, b)$. Por lo tanto,

$$b_{\mathcal{M}}(p, q) \leq b(e_M, a) + b(e_M, b) = B.$$

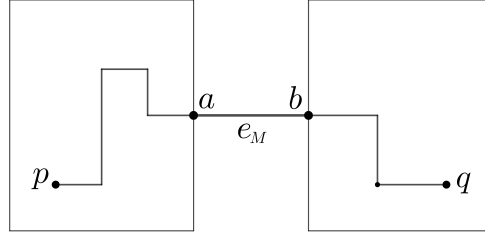


Figura 4.8: Configuración para el caso 1.

Caso 2: $p, q \in T[e_M, a]$, ver Figura 4.9. Sean $p = v_0, v_1, \dots, v_s = q$ los vértices de $P(p, q)$ en orden. Sea v_i el único vértice en esta trayectoria tal que $P(a, v_i)$ y $P(p, q)$ son disjuntos por aristas. Si $i = 0$ o $i = s$, entonces $b_{\mathcal{M}}(p, q) \leq b_{\mathcal{M}}(e_M, a) = b(e_M, a) < B$, por lo que suponemos que $1 \leq i \leq s - 1$, ver Figura 4.9b. Sin pérdida de generalidad, (v_i, v_{i+1}) forma un dobléz con la última arista de $P(a, v_i)$. Así, $b_{\mathcal{M}}(p, v_i) \leq b_{\mathcal{M}}(a, p) \leq b(e_M, a)$ y $b_{\mathcal{M}}(v_i, q) \leq b_{\mathcal{M}}(a, q) - 1 \leq b(e_M, a) - 1$. Por lo tanto,

$$b_{\mathcal{M}}(p, q) \leq b_{\mathcal{M}}(v_i, p) + b_{\mathcal{M}}(v_i, q) + 1 \leq 2b(e_M, a) \leq b(e_M, a) + b(e_M, b) \leq B.$$

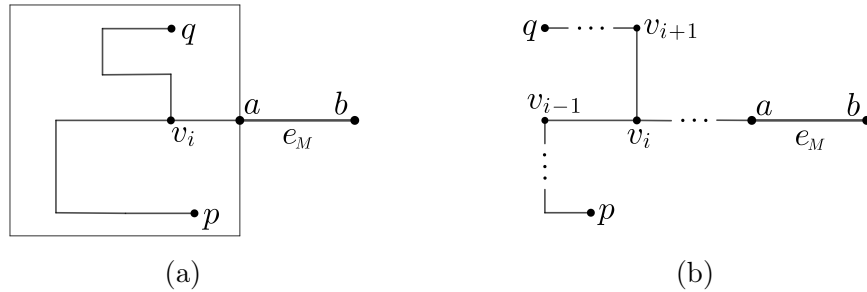


Figura 4.9: Configuración para el caso 2.

Caso 3. $p, q \in T[e_M, b]$, ver Figura 4.10. Si $b(e_M, a) = b(e_M, b)$, procedemos como en el Caso 2. De otro modo, por el Lema 4.4.4 se cumple $b(e, v) < b(e_M, b)$ si

$T[e, v] \subset T[e_M, b]$. Sean $p = v_0, v_1, \dots, v_s = q$ los vértices de $P(p, q)$ en orden. Sea v_i el único vértice tal que $P(b, v_i)$ y $P(p, q)$ son disjuntos por aristas. Como en el caso anterior, suponemos que $1 \leq i \leq s - 1$, ver Figura 4.10b. Sean $e_1 = (v_{i-1}, v_i)$ y $e_2 = (v_i, v_{i+1})$. Notemos que $T[e_1, v_{i-1}]$ y $T[e_2, v_{i+1}]$ están contenidos en $T[e_M, b]$, por lo que $b(e_1, v_{i-1})$ y $b(e_2, v_{i+1})$ son menores a $b(e_M, b)$. Como \mathcal{M} es óptimo respecto a $b(e_M, b)$, también lo es respecto a $b(e_1, v_{i-1})$ y $b(e_2, v_{i+1})$. Por lo tanto,

$$b_{\mathcal{M}}(p, q) \leq b_{\mathcal{M}}(p, v_i) + b_{\mathcal{M}}(v_i, q) + 1 \leq b(e_1, v_{i-1}) + b(e_2, v_{i+1}) + 1 \leq 2b(e_M, b) - 1 \leq B.$$

□

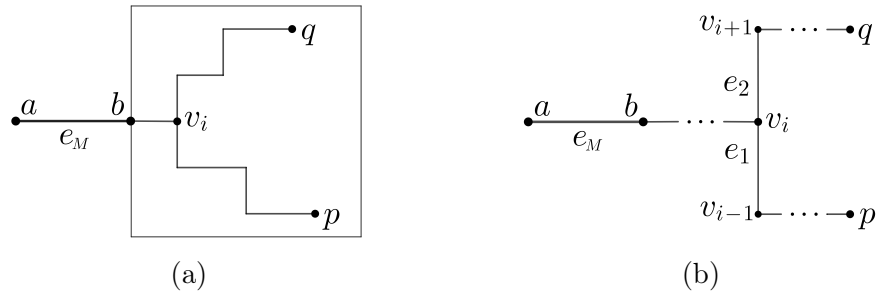


Figura 4.10: Configuración para el caso 3.

Notemos que los resultados presentados en este capítulo se extienden casi textualmente a árboles de grados mayores en dimensiones más altas. Es decir que si d está fija, podemos construir un encaje con segmentos ortogonales en $R^{\lceil d/2 \rceil}$ que minimice los dobleces de las trayectorias de un árbol de grado máximo d . Por otro lado, si proyectamos sobre el plano un encaje en una malla de dimensión alta, desde un ángulo adecuado, y posiblemente cambiando la longitud de algunas de las aristas del árbol de forma adecuada, podemos obtener un dibujo de T en el plano que requiere a lo más $\lceil d/2 \rceil$ orientaciones para representar a las aristas, lo que nos da el siguiente resultado:

Teorema 4.4.6. *Sea d un entero positivo. Todo árbol de grado máximo d puede ser encajado en el plano utilizando a lo más $\lceil d/2 \rceil$ pendientes para representar las aristas, de modo tal que se minimice el número de dobleces de la trayectoria que más se dobla.*

4.5. Área del encaje

En el trabajo presentado en este capítulo no ha sido una de nuestras prioridades reducir el área utilizada para dibujar a T . Sin embargo, podemos hacer las siguientes observaciones. Notemos que cualquier columna o fila de la malla en la que no se haya colocado un vértice pueden ser eliminadas sin alterar el número de dobleces de cualquier trayectoria. Esto, aunado al hecho de que cada fila o columna en la que se hayan dibujado k aristas debe tener al menos $k+1$ vértices, nos muestra que existe un encaje de T de $b(T)$ dobleces tal que todos los vértices yacen dentro de un rectángulo de tamaño $a \times b$ donde $a + 1 \leq n + 1$. Esta cota es justa para algunos árboles, como trayectorias y 4-estrellas donde cada arista es subdividida en cualquier número de aristas de menor longitud, véase la Figura 4.11a. Aún para árboles sin vértices de grado dos podríamos requerir espacio cuadrático, se muestra un ejemplo de esto en la Figura 4.11b. Estas observaciones contrastan con los resultados dados en [30], en donde se muestra que los árboles ternarios pueden ser encajados en mallas de área de tamaño menor a cuadrático. Sin embargo, es posible que se puedan obtener mejores cotas en esta dirección bajo la hipótesis de que $b(T)$ sea lo suficientemente grande.

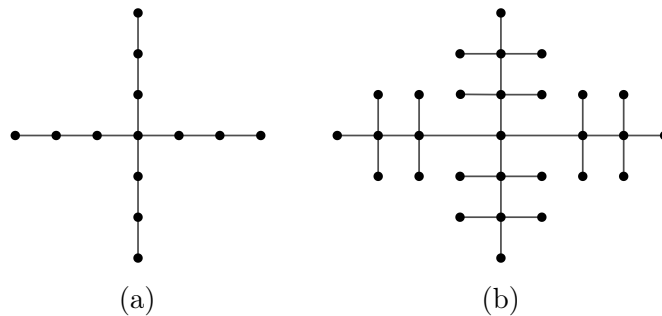


Figura 4.11: (a) Una 4-estrella cuyas aristas han sido subdivididas. (b) Un árbol sin vértices de grado dos, el cual requiere área cuadrática para ser dibujado en la malla entera cuando queremos minimizar $b(\mathcal{M})$.

Capítulo 5

Cuadriláteros arcoíris en conjuntos de puntos coloreados

5.1. Introducción

Un conjunto S de n puntos es k -coloreado si cada uno de sus elementos tiene asignado un color c_i , $1 \leq i \leq k$. En adelante suponemos que para cada i existe al menos un elemento de S con color c_i . Un polígono \mathcal{P} cuyos vértices son puntos en S es un *polígono arcoíris* si todos sus vértices tienen colores distintos; nos referiremos a \mathcal{P} como un polígono arcoíris de S . El área de un polígono simple \mathcal{P} es el área de la región encerrada por su frontera $\partial\mathcal{P}$. Un polígono cuyos vértices son puntos en S es *vacío* si no contiene elementos de S en su interior.

En este capítulo estudiamos dos problemas sobre polígonos arcoíris en conjuntos de puntos coloreados. El primer problema es encontrar un cuadrilátero arcoíris de área mínima o máxima en un conjunto de puntos k -coloreado S . Presentamos un algoritmo de tiempo $O(kn^2)$ para este problema. Notemos que un cuadrilátero arcoíris de área mínima en S no es necesariamente convexo o vacío, ver Figura 5.1. El segundo problema consiste en decidir si un conjunto de puntos 4-coloreado contiene un cuadrilátero arcoíris vacío, ya sea convexo o no. Para este problema presentamos un algoritmo de tiempo $O(n^3)$, el cual regresa un cuadrilátero arcoíris vacío convexo si este existe, y en caso contrario busca un cuadrilátero arcoíris no convexo.

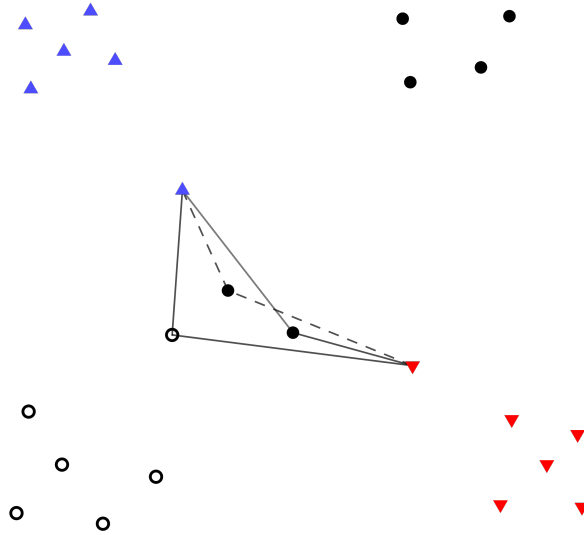


Figura 5.1: Un conjunto de puntos en el cual su único cuadrilátero arcoíris de área mínima no es convexo ni vacío.

5.1.1. Definiciones

Comenzamos describiendo la transformación dual entre puntos y rectas en el plano, y las propiedades de la subdivisión del plano inducida por un conjunto de rectas. Estos conceptos fueron estudiados ampliamente en [23] y [46], donde se utilizaron para encontrar triángulos de menor área en un conjunto de puntos en el plano entre otras aplicaciones.

Sea T la transformación geométrica definida como sigue: un punto dado $p = (a, b)$ es mapeado a una recta $\ell(p) : y = ax - b$, y una recta dada $\ell : y = mx + b$ es mapeada a un punto $p(\ell) = (m, -b)$. La transformación T preserva la incidencia, es decir, $p \in \ell$ si y sólo si $p(\ell) \in \ell(p)$. La transformación T también preserva el orden, es decir, p está arriba, sobre, o abajo de ℓ si y sólo si $p(\ell)$ está abajo, sobre, o arriba de $\ell(p)$, respectivamente.

Sea H un conjunto de n rectas en el plano. El conjunto H induce una subdivisión en el plano, la cual consiste en un conjunto de vértices, aristas y caras determinados por las intersecciones de los elementos de H . Esta subdivisión, la cual denotamos con $A(H)$, es conocido como el *arreglo de líneas* inducido por H . Algunas de las caras y aristas de $A(H)$ no están acotadas, es decir, se extienden hacia el infinito. Para lidiar con las regiones no acotadas, podemos suponer que el arreglo $A(H)$ está contenido

en rectángulo ortogonal lo suficientemente grande para contener todos los vértices de $A(H)$. Este rectángulo es conocido como la *caja contenedora* de $A(H)$.

Un arreglo de líneas $A(H)$ es *simple* si no existen dos rectas que sean paralelas y si no existen tres rectas concurrentes. La complejidad combinatoria de $A(H)$, que consiste en la suma del número vértices, aristas y caras de $A(H)$, es de $O(n^2)$. Se define la *zona* de una recta $\ell \in A(H)$ como el conjunto de caras de $A(H)$ cuya frontera es intersectado por ℓ . La complejidad de la zona de una recta $\ell \in A(H)$ es de $O(n)$. Los arreglos de líneas pueden ser construidos en tiempo $O(n^2)$ utilizando espacio de complejidad $O(n^2)$.

Dado un arreglo de líneas tal que el conjunto de rectas H que lo induce es el conjunto de rectas obtenidas al aplicar la transformación dual sobre conjunto de puntos S , es posible obtener el orden angular de los elementos de S alrededor de uno de sus puntos en tiempo $O(n)$ [8, 57]. Esto ocurre porque, dado un punto $p \in S$ y su recta dual $\ell(p) \in A(H)$, el orden de las intersecciones de $\ell(p)$ con las rectas de $A(H)$ es el orden angular de los puntos de S alrededor de p correspondientes a tales rectas.

5.2. Trabajo previo

Desde la década de 1980 diversos autores han estudiado problemas relacionados con encontrar polígonos cuyos vértices sean elementos de un conjunto de puntos en el plano y que cumplan con cierta propiedad, por ejemplo, que el área de la región que encierran sea mínima o máxima, ser convexo, o no contener otros elementos del conjunto de puntos en su interior. A continuación, mencionamos los trabajos más relevantes con respecto a los problemas estudiados en este capítulo, comenzando con aquellos relacionados a buscar polígonos cuya área sea mínima.

Chazelle et al. [23] y Edelsbrunner et al. [46] estudiaron de forma independiente el problema de encontrar un triángulo de área mínima cuyos vértices son puntos de un conjunto de cardinalidad n y presentaron algoritmos que utilizan tiempo y espacio $O(n^2)$. Ambos resultados son aplicaciones de una transformación dual entre rectas y puntos, así como de las propiedades de los arreglos de rectas. El espacio utilizado puede ser reducido si se evita tener que almacenar un número cuadrático de intersecciones de rectas en el arreglo. Edelsbrunner y Guibas [45] demostraron que una forma de lograr esto con espacio $O(n)$ es mediante una técnica conocida como *barrido topológico*. Para polígonos con un mayor número de lados, Eppstein et al. [48] demostraron que se pueden encontrar m -ángulos convexos de área mínima cuyos vértices sean elementos de un conjunto de n puntos en tiempo $O(mn^3)$. Su

algoritmo utiliza espacio de tamaño $O(n)$ para $m = 4$ y $O(n^2)$ para $m \geq 5$. Poco tiempo después, Eppstein [47] presentó un algoritmo de tiempo $O(n^2 \log n + 2^{6m} n^2)$ para este problema, el cual consigue una cota mejor cuando m es constante. Si se busca un pentágono de área mínima que no sea necesariamente convexo, este puede ser encontrado en tiempo $O(nT(n))$, donde $T(n)$ es el número de triángulos vacíos en el conjunto de puntos, lo cual fue demostrado por Hêche y Liebling [62].

Consideremos ahora los problemas relacionados con buscar m -ágonos de área máxima cuyos vértices son puntos de un conjunto P . Cordes [29] demostró que los vértices de un m -ágono de área máxima estarán en el cierre convexo de P siempre que se cumpla que m sea menor o igual a la cantidad de puntos en el cierre convexo de P . Lo anterior implica que si m es suficientemente pequeña o el cierre convexo de P contiene un número suficientemente grande de puntos este problema es equivalente a encontrar el m -ágono de área máxima cuyos vértices son puntos de un polígono convexo.

En 1979, Dobkin et al. [43] presentaron un algoritmo de tiempo lineal para encontrar un triángulo o un cuadrilátero de área máxima inscrito en un polígono convexo. Diversos autores se basaron en estos resultados para estudiar problemas relacionados a buscar m -ágonos de área máxima, sin embargo, un artículo reciente de Keikha et al. [102] muestra que el algoritmo de Dobkin et al. y los resultados basados en él son incorrectos. Esto hace relevante el resultado de Keikha et al. a pesar de que, desde 1992, existe un algoritmo paralelizable de tiempo lineal para encontrar un triángulo de área máxima en un polígono convexo, el cual fue presentado por Chandran y Mount [20]. Después de la publicación de Keikha et al. [102] nuevos algoritmos de tiempo lineal para encontrar el triángulo de área máxima en un polígono convexo fueron presentados de forma independiente por Jin [65] y por Kallus [66]. Por otro lado, Rote [83] probó que también es posible encontrar un cuadrilátero de área máxima inscrito en un polígono convexo en tiempo lineal.

Los problemas estudiados en este capítulo se encuentran en la intersección entre los problemas discutidos anteriormente y aquellos que tratan de encontrar cierto tipo de polígonos en conjuntos de puntos coloreados, como los que mencionamos a continuación. Cravioto et al. [31] demostraron que si S es un conjunto de puntos k -coloreado de tamaño suficientemente grande, entonces existe un triángulo monocromático con vértices en S que contiene a lo más $k - 3$ puntos en su interior para $k \geq 4$, además, probaron que siempre existe un cuadrilátero monocromático convexo con a lo más $2k - 3$ puntos en su interior para $k \geq 2$.

Un subconjunto de un conjunto de puntos S es llamado una *isla* si es la intersec-

ción de S con un conjunto convexo. Bautista et al. [11] presentaron un algoritmo de tiempo $O(n^3)$ para encontrar una isla monocromática de cardinalidad máxima en un conjunto de puntos 2-coloreado. Este mismo algoritmo, con modificaciones menores, puede resolver una versión del mismo problema en el que se han asignado pesos a los elementos de S y se busca una isla de peso máximo.

Se dice que un conjunto de puntos k -coloreado está balanceado si contiene el mismo número de puntos de cada color; un polígono está balanceado si su conjunto de vértices es un conjunto balanceado. Aichholzer et al. [1] probaron que cualquier conjunto de puntos 2-coloreado y balanceado con $2n$ elementos contiene al menos $1/45n^2 - \Theta(n)$ polígonos balanceados y vacíos de 6 vértices. Bereg et al. [13] demostraron que cualquier conjunto de puntos 2-coloreado y balanceado con $2n$ elementos contiene al menos $(n^2 - 4n)/12$ cuadriláteros balanceados vacíos y que esta cota es justa salvo por un factor constante. En el mismo artículo dieron una caracterización de los conjuntos de puntos 2-coloreados que contienen cuadriláteros vacíos.

El *índice arcoíris* r_k es el menor entero m tal que cada conjunto de puntos k -coloreado S contiene un polígono arcoíris *perfecto* con a lo más m vértices (un polígono arcoíris es perfecto si contiene exactamente un punto de cada color, ya sea en su frontera o en su interior). Flores-Peñaloza et al. [53] determinaron el valor de r_k para $k \leq 7$ y probaron que, en general, $(20n - 28)/19 \leq r_k \leq (10n)/7 + 11$.

Recientemente, Fabila-Monroy et al. [50] demostraron que cualquier conjunto de puntos k -coloreado en el cual cada clase cromática tenga al menos k puntos contiene $\Theta(k^3)$ triángulos arcoíris vacíos. Además, mostraron ejemplos de conjuntos de puntos que no contienen un cuadrilátero arcoíris vacío, ya sea convexo o no.

5.3. Cuadriláteros arcoíris de área mínima o máxima

El algoritmo presentado en esta sección se basa en el hecho de que todo cuadrilátero contiene al menos una *diagonal*, es decir, un segmento que une a dos de sus vértices y que está completamente contenido en el cuadrilátero. Hacemos la siguiente observación respecto a las diagonales en cuadriláteros arcoíris de área mínima.

Sea S un conjunto k coloreado y sea Q un cuadrilátero arcoíris de menor área contenido en S . Consideremos una diagonal $s(a, b)$ de Q . Sea $A = \triangle abq$ el triángulo arcoíris de área mínima tal que $q \in S$ está arriba de $\ell(a, b)$ y sea $B = \triangle abr$ el triángulo arcoíris de menor área tal que r está abajo de $\ell(a, b)$.

Observación 5.3.1. Si q y r tienen colores distintos, entonces el área de Q es igual al área de $A \cup B$.

Supongamos que los puntos q y r tienen el mismo color c_i . Sea C el triángulo arcoíris de área mínima con vértices a , b y un punto de color distinto a c_i arriba de $\ell(a, b)$. Definimos D de forma análoga con el tercer punto debajo de $\ell(a, b)$. De lo anterior podemos hacer la siguiente observación:

Observación 5.3.2. El área de Q es igual al área de $A \cup D$ o al área de $B \cup C$.

Las dos observaciones anteriores nos permiten utilizar la siguiente estrategia para encontrar el cuadrilátero arcoíris de área mínima.

Para cada par $\{a, b\}$ de puntos de color distinto en S buscamos el cuadrilátero arcoíris de menor área que tiene al segmento $s(a, b)$ como diagonal y posteriormente elegimos el cuadrilátero de menor área entre todos los resultantes.

Primero obtenemos, para cada color c_i distinto del color de a y del color de b , el triángulo arcoíris de área mínima Δabq tal que q tiene color c_i y está arriba de $\ell(a, b)$. Por las observaciones 5.3.1 y 5.3.2, es suficiente con almacenar dos triángulos arcoíris para cada diagonal bicromática $s(a, b)$ los triángulos Δabq y Δabr tales que:

- q y r tienen color distinto entre sí,
- para cualquier punto s del mismo color que q (respectivamente, s) arriba de $s(a, b)$, el área de Δabq (respectivamente, Δabr) es menor o igual que el área de Δabs ,
- para cualquier punto s arriba de $\ell(a, b)$ y de color distinto al de q o r se cumple que el área de Δabs es mayor o igual que la de Δabq y de Δabr .

De forma análoga, almacenamos dos triángulos arcoíris de área mínima con vértices a , b y un tercer punto de color distinto debajo de $\ell(a, b)$.

Una vez que hemos obtenido los cuatro triángulos arcoíris para $s(a, b)$ los usamos para encontrar el par de triángulos, uno de cada lado de $s(a, b)$, cuya unión es un cuadrilátero arcoíris de área mínima que contiene a $s(a, b)$ como una diagonal. Al proceder de manera similar con cada segmento bicromático en S obtenemos el cuadrilátero arcoíris de área mínima en S . Existen $O(n^2)$ pares de puntos de distinto color en S . Utilizaremos la transformación dual T como preprocesamiento de manera que los cuatro triángulos que almacenamos para cada segmento sean encontrados de manera eficiente.

Existe un resultado clásico que dice que un triángulo de área mínima cuyos vértices son puntos en un conjunto de puntos no coloreados $P \in \mathbb{R}^2$ puede ser encontrado en tiempo $O(n^2)$ utilizando la transformación dual T , ver [23] y [46]. A continuación, damos una breve descripción de su demostración.

Supongamos que un triángulo de área mínima en P tiene vértices $p, q, r \in P$. Entonces se cumple que r es el punto más cercano a la recta $\ell(p, q)$ entre todos los puntos en $P \setminus \{p, q\}$. Sea H el conjunto de líneas obtenido al aplicar la transformación T en los elementos de P y sea $A(H)$ el arreglo de líneas inducido por H . Entonces $\ell(p, q)$ se transforma en el punto $s = \ell(p) \cap \ell(q)$ y r se transforma en la recta $\ell(r)$, inmediatamente arriba o abajo de s verticalmente. Lo anterior se cumple ya que el punto contenido en $\ell(r)$ cuya coordenada x es igual que la coordenada x de s corresponde a la recta paralela a $\ell(p, q)$ que contiene a r . Para obtener el triángulo de área mínima que tiene como vértice al punto $r \in P$ se recorre la recta $\ell(r) \in A(H)$ mientras se obtiene, para cada vértice $v = \ell(p) \cap \ell(q)$ en cada cara de $A(H)$ incidente a $\ell(r)$, el área del triángulo Δpqr .

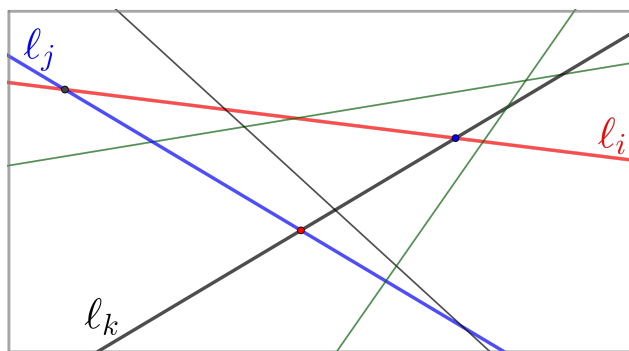


Figura 5.2: Ninguna de las tres rectas l_i, l_j, l_k interseca una cara que contenga la intersección de las otras dos rectas.

Consideremos ahora el problema de encontrar un triángulo arcoíris de menor área en un conjunto de puntos k -coloreado S . Sean p, q y r los vértices de un triángulo arcoíris de menor área en S . Sea H el conjunto de rectas duales de los puntos de S . La recta dual de p , $\ell(p)$, puede no intersecar ninguna cara de $A(H)$ que contenga el vértice $\ell(p) \cap \ell(r)$, ver Figura 5.2. Lo anterior también puede ocurrir para los puntos q y r , por lo cual la estrategia clásica podría fallar para encontrar el triángulo de área mínima que contenga los dos puntos de algún segmento bicromático. Sin embargo, si consideramos el arreglo $A(H_r)$ inducido por todas las rectas que tengan el mismo color que $\ell(r)$, y en dicho arreglo insertamos las rectas $\ell(p)$ y $\ell(q)$, entonces $\ell(r)$

contiene una arista de $A(H_r)$ contenida en la misma cara que $\ell(p) \cap \ell(q)$. Este hecho sugiere el algoritmo presentado a continuación.

5.3.1. Algoritmo

Sea H el conjunto de rectas duales de los elementos en S y sea $A(H)$ el arreglo de rectas inducido por H . Sea $A_i(H)$ el arreglo de líneas de H correspondientes a los puntos con color c_i para $i = 1, \dots, k$.

Sea p un punto de color c_i y sea c_j un color distinto a c_i . La recta dual de p , $\ell(p)$, puede ser dividida en un conjunto de intervalos tales que cualesquiera dos puntos contenidos en el mismo intervalo tienen una misma recta de $A_j(H)$ inmediatamente arriba de ellos (verticalmente). Aquí podemos hacer dos observaciones:

- Pueden existir intervalos sin una recta de $A_j(H)$ arriba (abajo) de ellos, a saber, intervalos de $\ell(p)$ contenidos en la envoltura superior (inferior) de $A_j(H \cup \ell(p))$.
- Los extremos de los intervalos tienen al menos dos rectas de $A_j(H)$ inmediatamente arriba de ellos (verticalmente).

La subdivisión de $\ell(p)$ en intervalos puede ser obtenida al insertar $\ell(p)$ en $A_j(H)$ para después recorrer al mismo tiempo $\ell(p)$ y las aristas de las caras de la zona de $\ell(p)$ en $A_j(H)$ arriba de $\ell(p)$ de izquierda a derecha. De manera análoga, separamos $\ell(p)$ en intervalos correspondientes a las rectas de $A_j(H)$ inmediatamente abajo de $\ell(p)$. Describimos a continuación la manera de encontrar los intervalos para las rectas arriba de $\ell(p)$.

Para el primer intervalo existen tres casos. Si una recta de $A_j(H)$ interseca la arista izquierda de la caja contenedora de $A_j(H)$ inmediatamente arriba de su intersección con $\ell(p)$, entonces el primer intervalo corresponde a dicha recta. De otro modo, si la primera intersección de la arista superior de la caja contenedora de $A_j(H)$ corresponde a una recta con pendiente negativa, entonces el primer intervalo corresponde a dicha recta. En el caso restante el primer intervalo no tiene una recta de $A_j(H)$ asignada y el segundo intervalo comienza en la primera intersección de $\ell(p)$ con una recta de $A_j(H)$ (la cual tendrá pendiente positiva), a la que corresponderá ese segundo intervalo.

Sea $L \in A_j(H)$ la recta actual en el recorrido de la zona de $\ell(p)$ en $A_j(H)$, la cual corresponde a un intervalo con punto inicial $a \in \ell(p)$. A continuación, mostramos la manera encontrar el punto final de este intervalo.

- Si la siguiente recta en el recorrido es $\ell(p)$, entonces el intervalo correspondiente a L termina en $\ell(p) \cap L$. Este punto también es el inicio del siguiente intervalo. Continuamos con el recorrido de la siguiente cara iniciando en L . El siguiente intervalo corresponde a la primera recta tal que la proyección vertical de su intersección con la recta anterior en el recorrido no interseque a L , ver figuras 5.3a, 5.3b y 5.3c. Puede pasar para el siguiente intervalo no exista una recta de $A_j(H)$ arriba de él, ver Figura 5.3d.
- Si la siguiente recta en el recorrido es una recta L' de color c_j , entonces revisamos la región de $\ell(p)$ contenida entre a y la proyección vertical de $L \cap L'$ en $\ell(p)$, que denotamos con α . Se muestra un ejemplo en la Figura 5.3e. Si no existe una recta de $A_j(H)$ que interseque a $\ell(p)$ entre a y α , entonces el intervalo actual termina en α . De otro modo, tomamos la recta L'' que interseca a $\ell(p)$ entre a y α tal que su punto de intersección con $\ell(p)$ es el más cercano a a . En este caso, $\ell(p)$ es el final del intervalo actual y el principio del siguiente, que corresponde a la recta L'' , ver el punto b en la Figura 5.3f.

El extremo derecho del último intervalo se determina de manera análoga al extremo izquierdo del primer intervalo.

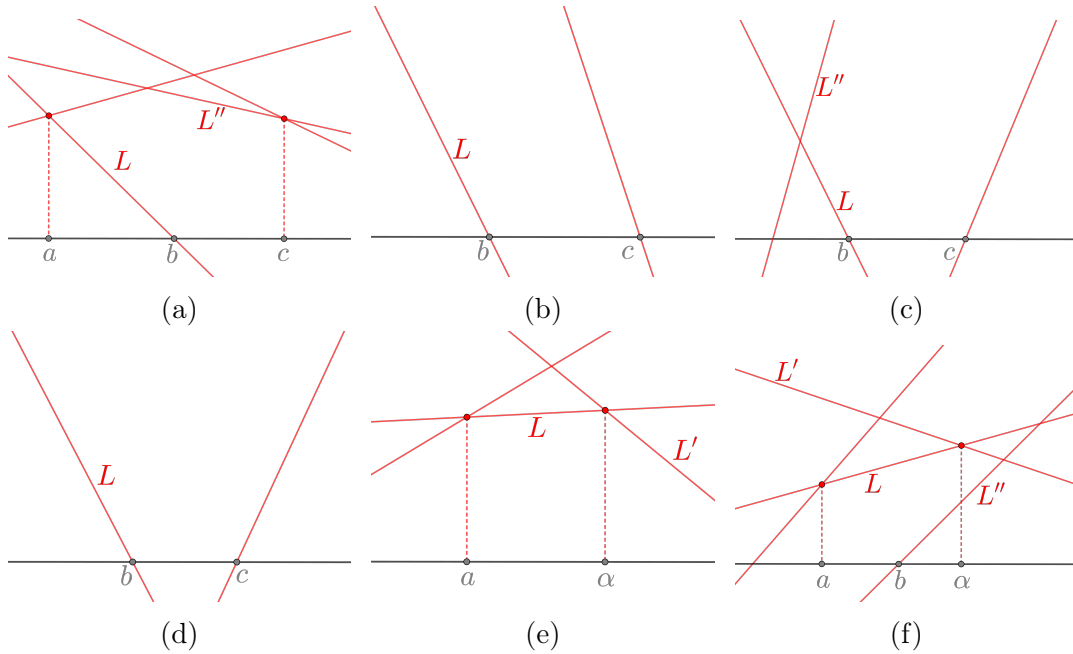


Figura 5.3: Posibles casos para el intervalo $[a, b]$.

Una vez que hemos obtenido el conjunto de intervalos correspondientes a las rectas de $A_j(H)$ inmediatamente arriba de $\ell(p)$ y el conjunto de intervalos correspondientes a las rectas de $A_j(H)$ inmediatamente abajo de $\ell(p)$, realizamos un recorrido de $\ell(p)$ de izquierda a derecha. Para cada recta $\ell(q)$ correspondiente a un punto q de un color distinto a c_i y c_j , obtenemos los triángulos Δpqp_j y $\Delta pqp'_j$, donde $\ell(p_j)$ es la recta de $A_j(H)$ arriba de $\ell(p)$ correspondiente al intervalo de $\ell(p)$ que contiene a $\ell(p) \cap \ell(q)$, y $\ell(p'_j)$ es la recta de $A_j(H)$ abajo de $\ell(p)$ correspondiente al intervalo que contiene a $\ell(p) \cap \ell(q)$. Finalizamos este paso al remover a $\ell(p)$ de $A_j(H)$.

Procedemos de esta manera para cada uno de los $(k-1)$ colores distintos al color de p . Mientras tanto, guardaremos para cada intersección de $\ell(p)$ con una recta q de color distinto. y para cada lado de $\ell(p)$, dos triángulos de área mínima Δabq y Δabr tal que q y r tienen color distinto.

En el último paso, recorreremos $\ell(p)$ una vez más y computamos para cada punto $\ell(p) \cap \ell(q)$ el cuadrilátero de área mínima que contiene la diagonal $s(p, q)$, el cual está dado por la unión de dos de los cuatro triángulos almacenados para $\ell(p) \cap \ell(q)$. Al repetir este paso para cada punto en S obtenemos el cuadrilátero arcoíris de área mínima cuyos vértices son puntos de S .

Ahora analizamos la complejidad del algoritmo. Construir $A(H)$ y $A_j(H)$ para $i = 1, \dots, k$ toma tiempo $O(k n^2)$. Cada recta $\ell(p) \in A_i(H)$ es insertada en $A_i(H)$ para $i = 1, \dots, k$. Luego recorreremos la zona de $\ell(p)$ en cada uno de esos arreglos para obtener los intervalos. Después, recorreremos $\ell(p)$ para obtener el cuadrilátero correspondiente a cada intersección de una recta con $\ell(p)$. Este paso toma tiempo $O(k n)$ para cada recta en $A(H)$. Por lo tanto, el tiempo de ejecución total del algoritmo es $O(k n^2)$ y tenemos el siguiente resultado:

Teorema 5.3.3. *Sea S un conjunto de puntos k -coloreado en \mathbb{R}^2 de cardinalidad n en posición general. Un cuadrilátero arcoíris de área mínima cuyos vértices son puntos en S puede ser encontrado en tiempo $O(k n^2)$ usando espacio $O(n^2)$.*

Observemos que el proceso anterior tiene como paso intermedio encontrar el triángulo arcoíris de área mínima que tiene como vértices los extremos de cada segmento bicromático en S . Así, el siguiente corolario se sigue directamente de la prueba del Teorema 5.3.3:

Corolario 5.3.4. *Sea S un conjunto de puntos k -coloreado en \mathbb{R}^2 de cardinalidad n en posición general. Un triángulo arcoíris de área mínima cuyos vértices son puntos en S puede ser encontrado en tiempo $O(k n^2)$ usando espacio $O(n^2)$.*

Consideremos ahora el problema de encontrar un cuadrilátero arcoíris de área máxima cuyos vértices sean puntos en S . Podemos seguir un razonamiento similar al de la prueba del Teorema 5.3.3, y obtener, para cada recta $\ell(p) \in A(H)$, un conjunto de intervalos a partir de las rectas de cada color más alejadas verticalmente arriba y abajo de $\ell(p)$. Notemos que, como en el caso anterior, pueden existir intervalos de $\ell(p)$ para los que no haya una recta abajo o para los que no haya una recta arriba de ellos. Estos intervalos corresponden a secciones de $\ell(p)$ contenidas en la envoltura inferior o superior o $A(H)$. Las rectas correspondientes a los siguientes intervalos son obtenidas con recorridos similares a los anteriores. Por lo tanto, tenemos los siguientes resultados:

Teorema 5.3.5. *Sea S un conjunto de puntos k -coloreado en \mathbb{R}^2 de cardinalidad n en posición general. Un cuadrilátero arcoíris de área máxima cuyos vértices son puntos en S puede ser encontrado en tiempo $O(kn^2)$ usando espacio $O(n^2)$.*

Corolario 5.3.6. *Sea S un conjunto de puntos k -coloreado en \mathbb{R}^2 de cardinalidad n en posición general. Un triángulo arcoíris de área máxima cuyos vértices son puntos en S puede ser encontrado en tiempo $O(kn^2)$ usando espacio $O(n^2)$.*

5.4. Cuadriláteros arcoíris vacíos

En esta sección estudiamos el problema de decidir si, dado un conjunto de n puntos 4-coloreado S , existe un cuadrilátero arcoíris vacío, ya sea convexo o no, cuyos vértices sean puntos de S . El resultado que presentamos es un algoritmo de tiempo $O(n^3)$ que reporta un cuadrilátero arcoíris vacío y convexo en S , si este existe. En caso contrario, el algoritmo reporta, si existe, un cuadrilátero arcoíris vacío que no sea convexo cuyos vértices sean puntos en S .

Sean p y q puntos en S con diferente color y supongamos sin pérdida de generalidad que p tiene color c_1 , q tiene color c_2 , $\ell(p, q)$ es horizontal, y p está a la izquierda de q . Vamos a encontrar, si existe, un cuadrilátero arcoíris vacío Q tal que $s(pq)$ es una diagonal de Q . Para esto, primero encontraremos todos los triángulos arcoíris vacíos cuyos vértices son p , q , y un punto arriba o abajo de $\ell(p, q)$.

Supongamos que un cuadrilátero arcoíris vacío Q existe en S . Sean r y s , respectivamente, los vértices de Q arriba y abajo de $\ell(p, q)$. Entonces, si Q es convexo, el punto s se encuentra en la región a la derecha de $\ell(r, q)$, a la izquierda de $\ell(r, p)$ y abajo de $\ell(p, q)$. Llamamos a esta región la *región factible* de r .

Los triángulos arcoíris vacíos que contienen a p , q y un punto arriba de $\ell(p, q)$ son encontrados de la siguiente manera. Sean C_1, C_2, C_3 , y C_4 los subconjuntos de puntos de S contenidos arriba de $\ell(p, q)$ que tienen color c_1, c_2, c_3 , y c_4 , respectivamente. Sea $S'' = C_1 \cup C_2 \cup C_3 \cup C_4$. Obtenemos el orden radial en sentido de los elementos de S'' , en sentido horario alrededor de p comenzando desde q , y en sentido antihorario alrededor de q comenzando desde p . Definimos un conjunto R que contiene, por cada elemento r en S'' , un punto $r' = (x_r, y_r)$ tal que x_r es la posición de r alrededor de p y y_r es la posición de r alrededor de q .

Ahora consideremos dos puntos r y s de S'' y supongamos que s está dentro del triángulo Δpqr . Entonces se cumple que s es mayor que r en ambos ordenamientos radiales, por lo que s' domina a r' ; se muestra un ejemplo de esto en la Figura 5.4. Por lo tanto, el triángulo Δpqr es un triángulo vacío si y sólo si r' es un punto maximal de R . Dado que los elementos de R ya están ordenados respecto a ambas coordenadas, podemos encontrar su subconjunto de elementos maximales (y sus puntos correspondientes S'') en tiempo $O(n)$. Después, de los puntos obtenidos conservamos solamente aquellos que tengan color c_3 o c_4 . Si no quedan puntos de un lado de $\ell(p, q)$, o si todos los puntos restantes son del mismo color entonces no existe un cuadrilátero arcoíris con $s(p, q)$ como diagonal. Supongamos que este no es el caso. Entonces los puntos almacenados aparecen en el mismo orden con respecto a p y q . Separamos estos puntos en dos subconjuntos: $\{a_1, \dots, a_t\}$ contiene los puntos que están arriba de $\ell(p, q)$ y $\{b_1, \dots, b_u\}$ contiene los puntos abajo de $\ell(p, q)$, ambos ordenados en sentido horario alrededor de p .

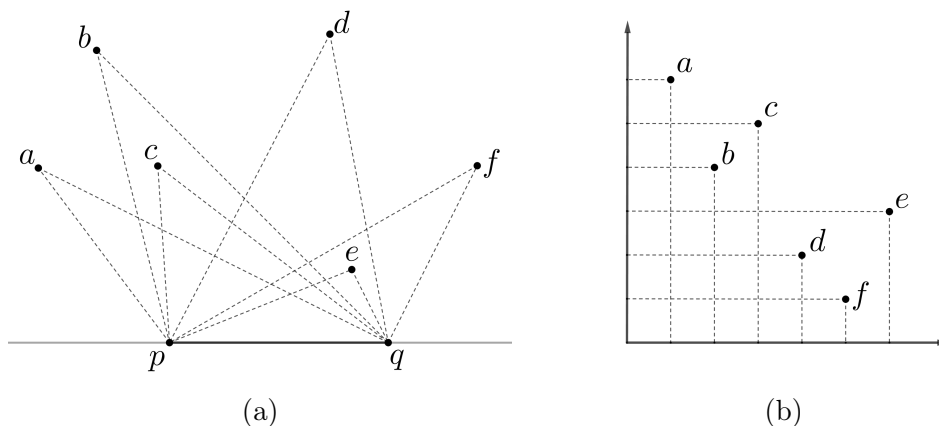


Figura 5.4: (a) El conjunto de puntos de S'' arriba de $\ell(p, q)$. (b) El conjunto de puntos R generado a partir de los órdenes alrededor de p y q .

Si existe al menos un par $a \in \{a_1, \dots, a_t\}$, $b \in \{b_1, \dots, b_u\}$ tal que a y b tienen colores diferentes, entonces recorremos los elementos de ambos conjuntos en orden dos veces; ver la Figura 5.5 para un ejemplo. En el primer recorrido consideramos solamente los puntos de $C_3 \cap \{a_1, \dots, a_t\}$ y $C_4 \cap \{b_1, \dots, b_u\}$, y en el segundo recorrido consideramos solamente los puntos de $C_4 \cap \{a_1, \dots, a_t\}$ y $C_3 \cap \{b_1, \dots, b_u\}$. Sean a_i y b_j los puntos actuales en el recorrido. Si b_j está a la izquierda de $\ell(a_i, q)$, entonces está fuera de la región factible de a_m , $m \geq i$. En este caso nos movemos al punto b_{j+1} . Del mismo modo, si b_j está a la derecha de $\ell(a_i, q)$ nos movemos al punto a_{i+1} . En el caso restante $\square pa_iqb_j$ es un cuadrilátero arcoíris vacío convexo. Si sucede que terminan los dos recorridos y no se ha encontrado un cuadrilátero convexo, regresamos $\square paqb$, el cual es un cuadrilátero arcoíris vacío.

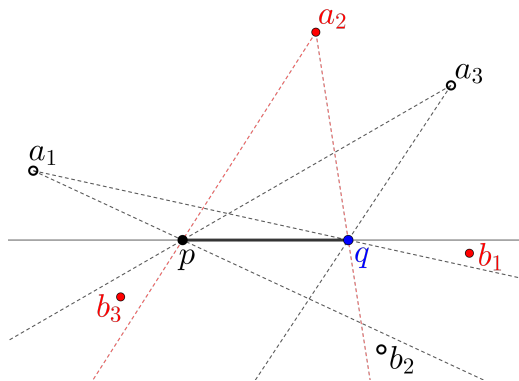


Figura 5.5: Cuando los puntos del recorrido son a_3 y b_3 encontramos un cuadrilátero arcoíris convexo vacío.

Al repetir el proceso anterior para cada par de puntos obtenemos el siguiente resultado:

Teorema 5.4.1. *Podemos decidir si S contiene un cuadrilátero arcoíris convexo vacío o un cuadrilátero arcoíris vacío en tiempo $O(n^3)$ y usando espacio $O(n^2)$.*

Capítulo 6

Conclusiones

A continuación hacemos una breve recopilación de las contribuciones presentadas en cada capítulo de esta tesis, así como del posible trabajo a futuro en los problemas que estudiamos:

En el Capítulo 2 estudiamos una variante del problema de la galería de arte inspirada en aplicaciones de drones (también conocidos como vehículos aéreos no tripulados). En este problema queremos colocar el número mínimo de estaciones de carga necesarias sobre el contorno de una isla \mathcal{I} de perímetro L , de manera que un dron con rango de vuelo d pueda darle la vuelta completa a \mathcal{I} . Los drones pueden volar sobre el contorno de \mathcal{I} o sobre el mar, y usan dichas estaciones para reabastecer su energía. Presentamos un algoritmo de tiempo $O(n^2 + \frac{L}{d}n)$ que regresa una solución óptima, o con una estación adicional con respecto a una solución óptima. En caso de que se requiera una estación colocada en el cierre convexo de la isla, la solución regresada por el algoritmo es óptima.

Entre la literatura sobre aplicaciones de drones disponible actualmente podemos encontrar varios tipos de resultados. Algunos son del tipo teórico, tal es el caso del resultado presentado en esta tesis. Por otro lado, existen artículos en los que, en lugar de buscar una solución exacta, se utilizan heurísticas que permitan obtener soluciones aceptables para problemas en los que participan un número considerablemente grande de factores. Los resultados aquí presentados podrían utilizarse para avanzar en la segunda dirección, ya que nuestro algoritmo permite cierta adaptabilidad para tomar en cuenta otros factores.

Nuestra suposición de que los drones vuelan alrededor del contorno de una isla nos

permitió modelar este problema como uno de polígonos en \mathbb{R}^2 . Así, pudimos asumir que todo dron volará a una altura constante durante su recorrido, lo que vuelve razonable la suposición de que los drones siempre podrán volar una distancia d entre recargas. Sin embargo, en otras aplicaciones podría suceder que la distancia máxima que un dron puede desplazarse entre dos estaciones varíe debido a factores como cambios de elevación del suelo o la presencia de obstáculos. En estos casos podría ser necesario computar esta distancia cada vez que se coloque una nueva estación. Si fuera posible realizar estos cálculos en tiempo lineal cada vez, nuestro algoritmo podría adaptarse para enfrentar estos casos sin aumentar la complejidad temporal. Para lograr esto basta con cambiar el valor de d en los pasos que se describen en los lemas 2.5.7 y 2.5.8; estos dos pasos se realizan para cada estación colocada y toman tiempo lineal.

Por otro lado, el algoritmo aquí presentado podría utilizarse como subrutina en algoritmos destinados a resolver otros problemas sobre drones. Un ejemplo de esto es un problema que estudiamos en el artículo que enviamos a *Theoretical Computer Science* y que no fue incluido en esta tesis. En dicho problema tenemos permitido colocar k estaciones de recarga en el contorno de la isla, y queremos colocarlas de modo que se minimice el rango de vuelo d necesario para que los drones puedan realizar su recorrido. Diseñamos un algoritmo que regresa una aproximación aditiva para este rango de vuelo, el cual utiliza como subrutina el algoritmo presentado en esta tesis.

Por último, el trabajo a futuro más evidente que se desprende de este resultado es determinar si el problema de obtener la solución óptima para el número de estaciones (sin restricciones) es polinomial y, en el caso positivo, diseñar un algoritmo que obtenga dicha solución.

En el Capítulo 3 presentamos un resultado que es continuación del trabajo realizado durante los estudios de maestría del autor. El problema estudiado en este capítulo es el de *cobertura con balizas*, una variante del problema de la galería de arte en la que los guardias, en lugar de vigilar aquellos puntos que les sean visibles, vigilan todos aquellos puntos que puedan atraer hacia sí mismos como lo haría un imán con una partícula de hierro. Aquí, estudiamos el problema de cobertura con balizas en una familia de poliedros ortogonales llamados *ortoárboles*. Además, añadimos la restricción de que las balizas sean colocadas sobre vértices de estos poliedros.

En este capítulo probamos que $\lfloor n/12 \rfloor$ balizas son siempre suficientes para vigilar un ortoárbol de n vértices. Con este resultado obtenemos la cota combinatoria justa,

ya que la cota inferior se presentó en la tesis de maestría del autor, ver [74]. Dicha cota inferior se incluye también en esta tesis con la intención de mostrar el resultado completo. El resultado que presentamos concluye con esta línea de investigación, por lo que el posible trabajo futuro relacionado con este problema comienza con la búsqueda de otras familias no triviales de poliedros ortogonales que puedan ser vigiladas con balizas colocadas en sus vértices.

En el Capítulo 4 estudiamos un problema acerca de encontrar encajes ortogonales de gráficas en los que las aristas de las gráficas sean representadas con segmentos de recta, y sus vértices con puntos de coordenadas enteras. Presentamos un algoritmo de tiempo lineal que, dado un árbol T de grado máximo cuatro, encuentra un encaje ortogonal de T que minimiza la máxima cantidad de dobleces entre sus trayectorias.

En las últimas décadas se han estudiado muchos problemas relacionados a encontrar encajes rectilíneos de gráficas, por lo que nos resultó sorprendente que esta variante no hubiese sido estudiada aún. Por esta razón, podría resultar de interés agregar otras restricciones al encaje buscado. Por ejemplo, podríamos tratar de minimizar el área de su caja contenedora: Como mencionamos anteriormente, existen casos en los que el encaje que minimiza el máximo número de dobleces por trayectoria, $b(T)$, requiere área cuadrática. Sin embargo, este podría no ser el caso si $b(T)$ es lo suficientemente grande, por lo que una posible línea de investigación para trabajo futuro sería que el área del encaje encontrado sea sensible al valor $b(T)$.

Continuando con la misma línea de investigación, se podrían considerar otras familias de gráficas que han sido estudiadas en la literatura sobre encajes rectilíneos de gráficas. Por ejemplo, se pueda decidir en tiempo polinomial si una gráfica plana-exterior o una gráfica en serie-paralela tiene un encaje ortogonal, por lo que es factible tratar de minimizar los dobleces de las trayectorias en este tipo de gráficas. Además, podemos considerar aquellas familias de gráficas para las que se ha determinado que se pueden dibujar con una cantidad constante de dobleces en sus aristas, pero ahora con el fin de minimizar el máximo número de dobleces de sus trayectorias. Dado que en estos tipos de gráficas puede existir más de una trayectoria uniendo un par de vértices, podemos considerar los dobleces de una trayectoria más corta entre cada par de vértices e ignorar el resto de las trayectorias que los unen.

Otra posible variante a considerar es aquella en la que los vértices de una gráfica son asignados a puntos fijos en el plano antes de buscar el encaje. En esta variante estamos obligados a permitir dobleces en las aristas si queremos evitar o reducir los cruces entre ellas. Así, las variantes en las cuales trabajar podrían tratar sobre

minimizar los dobleces en el interior de las aristas evitando por completo los cruces (ya sea el número máximo por arista o la suma total), o acerca de minimizar la suma total de cruces y dobleces en el encaje.

Finalmente, en el Capítulo 5 estudiamos dos problemas en puntos coloreados en el plano. En el primer problema, dado un conjunto S de n puntos coloreados con k colores, queremos encontrar el cuadrilátero arcoíris de área máxima o de área mínima cuyos vértices sean elementos de S . Presentamos un algoritmo de tiempo $O(kn^2)$ para solucionar este problema. El segundo problema trata sobre encontrar un cuadrilátero arcoíris vacío en un conjunto de n puntos coloreados con 4 colores. Presentamos un algoritmo de tiempo $O(n^3)$ que encuentra, si existe, un cuadrilátero arcoíris vacío y convexo. Si no existe, el algoritmo decide si existe un cuadrilátero arcoíris vacío y en caso positivo lo reporta.

Para el primer problema podemos considerar el siguiente trabajo a futuro. Respecto a encontrar un cuadrilátero arcoíris de menor área, reducir el factor n^2 de forma significativa parece poco probable, dado que se sabe que encontrar el triángulo de menor área es un problema 3SUM-difícil (este problema es evidentemente más difícil que encontrar tripletas de puntos alineados en el plano). Por lo tanto, consideramos como el principal problema abierto reducir el factor k , posiblemente para obtener un algoritmo de tiempo $O(n^2 \log k)$. El caso de los cuadriláteros de área máxima es distinto, puesto que se sabe que encontrar un triángulo de área máxima en un conjunto de puntos toma tiempo $O(n \log n)$ (el problema es lineal en polígonos convexos, por lo que requerimos primero construir el cierre convexo del conjunto, el cual define el triángulo de área máxima). Así, quedaría abierto el problema de determinar si se pueden reducir ambos factores en la complejidad. Para el segundo problema también creemos posible reducir la complejidad al menos a $O(n^2 \log n)$, sin embargo no hemos encontrado una manera de lograrlo a la fecha.

Otras posibles líneas de investigación podrían considerar polígonos arcoíris con un mayor número de lados, ya sea para un número de lados constante, como los pentágonos, o bien, encontrar una generalización para polígonos con m vértices.

Bibliografía

- [1] O. Aichholzer, J. Urrutia, and B. Vogtenhuber. Balanced 6-holes in bichromatic point sets. In *Japanese Conference on Discrete and Computational Geometry*, pages 5–6, 2013.
- [2] I. Aldana-Galván, C. Alegria, J. L. Álvarez-Rebollar, N. Marín, E. Solís-Villarreal, J. Urrutia, and C. Velarde. Finding minimum witness sets in orthogonal polygons. *Computational Geometry*, 90:101656, 2020.
- [3] I. Aldana-Galván, J. L. Álvarez-Rebollar, J. C. Catana-Salazar, M. Jiménez-Salinas, E. Solís-Villarreal, and J. Urrutia. Minimizing the solid angle sum of orthogonal polyhedra and guarding them with $\frac{\pi}{2}$ -edge guards. In *Proceedings of the 28th Canadian Conference on Computational Geometry, Vancouver, August 3-5*, pages 175–181, 2016.
- [4] I. Aldana-Galván, J. L. Álvarez-Rebollar, J. C. Catana-Salazar, N. Marín, E. Solís-Villarreal, J. Urrutia, and C. Velarde. Tight bounds for illuminating and covering of orthotrees with vertex lights and vertex beacons. *Graphs and Combinatorics*, 36(3):617–630, 2020.
- [5] I. Aldana-Galván, J. L. Álvarez-Rebollar, J. C. Catana-Salazar, N. Marín-Nevárez, E. Solís-Villarreal, J. Urrutia, and C. Velarde. Covering orthotrees with guards and beacons. In *In proceedings of XVII Spanish Meeting on Computational Geometry, Alicante, Spain, July 26-28*, pages 56–68, 2017.
- [6] J. Álvarez-Rebollar, J. Cravioto-Lagos, N. Marín, E. Solís-Villarreal, and J. Urrutia. Representing point sets on the plane as permutations. *Information Processing Letters*, 175:106228, 2022.
- [7] A. R. Arévalo Loyola. *Detección de cuadriláteros convexos heterocromáticos en conjuntos de puntos 4-coloreados*. Tesis de maestría, Universidad Nacional Autónoma de México, 2021.

- [8] T. Asano, T. Asano, L. Guibas, J. Hersberger, and H. Imai. Visibility of disjoint polygons. *Algorithmica*, 1(1-4):49–63, 1986.
- [9] S. W. Bae, C.-S. Shin, and A. Vigneron. Tight bounds for beacon-based coverage in simple rectilinear polygons. *Computational Geometry*, 80:40–52, 2019.
- [10] L. Bass and S. Schubert. On finding the disc of minimum radius containing a given set of points. *Mathematics of Computation*, 21(100):712–714, 1967.
- [11] C. Bautista-Santiago, J. M. Díaz-Báñez, D. Lara, P. Pérez-Lantero, J. Urrutia, and I. Ventura. Computing optimal islands. *Operations Research Letters*, 39(4):246–251, 2011.
- [12] N. M. Benbernou, E. D. Demaine, M. L. Demaine, A. Kurdia, J. O’Rourke, G. Toussaint, J. Urrutia, and G. Viglietta. Edge-guarding orthogonal polyhedra. In *Proceedings of the 23rd Canadian Conference on Computational Geometry, Toronto, August 10-12*, pages 461–466, 2011.
- [13] S. Bereg, J. M. Díaz-Báñez, R. Fabila-Monroy, P. Pérez-Lantero, A. Ramírez-Vigueras, T. Sakai, J. Urrutia, and I. Ventura. On balanced 4-holes in bichromatic point sets. *Computational Geometry*, 48(3):169–179, 2015.
- [14] B. Bhattacharya, M. Burmester, Y. Hu, E. Kranakis, Q. Shi, and A. Wiese. Optimal movement of mobile sensors for barrier coverage of a planar region. *Theoretical Computer Science*, 410(52):5515–5528, 2009.
- [15] M. Biro. *Beacon-based routing and guarding*. PhD thesis, State University of New York at Stony Brook, 2013.
- [16] M. Biro, J. Gao, J. B. Iwerks, I. Kostitsyna, and J. S. Mitchell. Beacon-based routing and coverage. In *21st Fall Workshop on Computational Geometry, New York, November 4-5*, 2011.
- [17] N. Boltyanskii and N. Yaglom. *Convex figures*. Holt, Rinehart and Winst., 1961.
- [18] S. Carlsson and H. Jonsson. Computing a shortest watchman path in a simple polygon in polynomial-time. In *Workshop on Algorithms and Data Structures*, pages 122–134. Springer, 1995.
- [19] T. M. Chan. Tree drawings revisited. *Discrete and Computational Geometry*, 63:799–820, 2020.

- [20] S. Chandran and D. M. Mount. A parallel algorithm for enclosed and enclosing triangles. *International Journal of Computational Geometry & Applications*, 2(02):191–214, 1992.
- [21] R. G. Chávez Jiménez. *Implementación de algoritmos en conjuntos de puntos 3, 4-coloreados*. Tesis de maestría, Universidad Nacional Autónoma de México, 2021.
- [22] B. Chazelle. Triangulating a simple polygon in linear time. *Discrete & Computational Geometry*, 6(3):485–524, 1991.
- [23] B. Chazelle, L. J. Guibas, and D.-T. Lee. The power of geometric duality. *BIT Numerical Mathematics*, 25(1):76–90, 1985.
- [24] W.-P. Chin and S. Ntafos. Optimum watchman routes. In *Proceedings of the second annual symposium on Computational geometry*, pages 24–33, 1986.
- [25] W.-P. Chin and S. Ntafos. Shortest watchman routes in simple polygons. *Discrete & Computational Geometry*, 6(1):9–31, 1991.
- [26] G. Chrystal. On the problem to construct the minimum circle enclosing n given points in the plane. *Proceedings of the Edinburgh Mathematical Society*, 3(1885):30–33, 1885.
- [27] V. Chvátal. A combinatorial theorem in plane geometry. *Journal of Combinatorial Theory*, 18(1):39–41, 1975.
- [28] J. Cleve. Combinatorics of beacon-based routing and guarding in three dimensions. *Master's thesis, Freie Universität Berlin*, 2017.
- [29] C. Cordes and K. Reid. Largest polygons with vertices in a given finite set. *Discrete applied mathematics*, 14(3):255–262, 1986.
- [30] B. Covella, F. Frati, and M. Patrignani. On the area requirements of planar straight-line orthogonal drawings of ternary trees. *Theoretical Computer Science*, 852:197 – 211, 2021.
- [31] J. Cravioto-Lagos, A. C. González-Martínez, T. Sakai, and J. Urrutia. On almost empty monochromatic triangles and convex quadrilaterals in colored point sets. *Graphs and Combinatorics*, 35(6):1475–1493, 2019.
- [32] P. Crescenzi, G. Di Battista, and A. Piperno. A note on optimal area algorithms for upward drawings of binary trees. *Computational Geometry*, 2(4):187–200, 1992.

- [33] J. Czyzowicz, P. Egyed, H. Everett, D. Rappaport, T. Shermer, D. Souvaine, G. Toussaint, and J. Urrutia. The aquarium keeper’s problem. In *Proceedings of the second annual ACM-SIAM symposium on Discrete algorithms*, pages 459–464, 1991.
- [34] M. Damian and R. Flatland. Unfolding low-degree orthotrees with constant refinement. In *30th Canadian Conference on Computational Geometry, Winnipeg, August 8-10*. Elsevier, 2018.
- [35] M. Damian and R. Flatland. Unfolding orthotrees with constant refinement. *arXiv preprint arXiv:1811.01842*, 2018.
- [36] H. de Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10(1):41 – 51, 1990.
- [37] V. T. F. de Luca, F. de Souza Oliveira, and J. L. Szwarcfiter. Minimum number of bends of paths of trees in a grid embedding. In *LAGOS 2021*, 2021.
- [38] B. Delaunay. Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7(793-800):1–2, 1934.
- [39] O. Devillers, F. Hurtado, G. Károlyi, and C. Seara. Chromatic variants of the erdos–szekeres theorem on points in convex position. *Computational Geometry*, 26(3):193–208, 2003.
- [40] G. Di Battista, G. Liotta, and F. Vargiu. Spirality and optimal orthogonal drawings. *SIAM Journal on Computing*, 27(6):1764–1811, 1998.
- [41] W. Didimo, M. Kaufmann, G. Liotta, and G. Ortali. Rectilinear planarity testing of plane series-parallel graphs in linear time. In *Graph Drawing, Symposium on Graph Drawing and Network Visualization*, pages 436–449, 2020.
- [42] E. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.
- [43] D. P. Dobkin and L. Snyder. On a general method for maximizing and minimizing among certain geometric problems. In *20th Annual Symposium on Foundations of Computer Science (sfcs 1979)*, pages 9–17. IEEE, 1979.
- [44] H. Edelsbrunner. *Algorithms in combinatorial geometry*, volume 10. Springer Science & Business Media, 1987.
- [45] H. Edelsbrunner and L. J. Guibas. Topologically sweeping an arrangement. *Journal of Computer and System Sciences*, 38(1):165–194, 1989.

- [46] H. Edelsbrunner, J. O'Rourke, and R. Seidel. Constructing arrangements of lines and hyperplanes with applications. *SIAM Journal on Computing*, 15(2):341–363, 1986.
- [47] D. Eppstein. New algorithms for minimum area k -gons. In *Proceedings of the third annual ACM-SIAM symposium on Discrete algorithms*, pages 83–88, 1992.
- [48] D. Eppstein, M. Overmars, G. Rote, and G. Woeginger. Finding minimum areak-gons. *Discrete & Computational Geometry*, 7(1):45–58, 1992.
- [49] P. Erdős and G. Szekeres. A combinatorial problem in geometry. *Compositio mathematica*, 2:463–470, 1935.
- [50] R. Fabila-Monroy, D. Perz, and A. L. Trujillo-Negrete. Empty rainbow triangles in k -colored point sets. *Computational Geometry*, 95:101731, 2021.
- [51] I. Fáry. On straight lines representation of planar graphs. *Acta Sci. Math. Szeged*, 11(4):229–233, 1948.
- [52] S. Fisk. A short proof of Chvátal's watchman theorem. *J. Combinatorial Theory (B)*, 24:374, 1978.
- [53] D. Flores-Peñaloza, M. Kano, L. Martínez-Sandoval, D. Orden, J. Tejel, C. D. Tóth, J. Urrutia, and B. Vogtenhuber. Rainbow polygons for colored point sets in the plane. *Discrete Mathematics*, 344(7):112406, 2021.
- [54] A. R. Forrest. *Curves and surfaces for computer-aided design*. PhD thesis, University of Cambridge, 1968.
- [55] F. Frati. Planar rectilinear drawings of outerplanar graphs in linear time. In *Graph Drawing, Symposium on Graph Drawing and Network Visualization*, pages 423–436, 2020.
- [56] A. Garg and R. Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM Journal on Computing*, 31(2):601–625, 2001.
- [57] S. K. Ghosh. *Visibility algorithms in the plane*. Cambridge university press, 2007.
- [58] S. K. Ghosh, A. Maheshwari, S. P. Pal, S. Saluja, and C. V. Madhavan. Characterizing and recognizing weak visibility polygons. *Computational Geometry*, 3(4):213–233, 1993.

- [59] R. L. Graham. An efficient algorithm for determining the convex hull of a finite planar set. *Information Processing Letters*, 1:132–133, 1972.
- [60] L. Guibas, J. Hershberger, D. Leven, M. Sharir, and R. E. Tarjan. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, 2(1):209–233, 1987.
- [61] M. Hammar and B. J. Nilsson. Concerning the time bounds of existing shortest watchman route algorithms. In *International Symposium on Fundamentals of Computation Theory*, pages 210–221. Springer, 1997.
- [62] J.-F. Hêche and T. M. Liebling. Finding minimum area simple pentagons. *Operations research letters*, 21(5):229–233, 1997.
- [63] E. Helly. Über systeme von abgeschlossenen mengen mit gemeinschaftlichen punkten. *Monatshefte für Mathematik und Physik*, 37(1):281–302, 1930.
- [64] R. Honsberger. Mathematical gems ii. *The Mathematical Gazette*, 61(417):227–228, 1977.
- [65] K. Jin. Maximal area triangles in a convex polygon. *arXiv preprint arXiv:1707.04071*, 2017.
- [66] Y. Kallus. A linear-time algorithm for the maximum-area inscribed triangle in a convex polygon. *arXiv preprint arXiv:1706.03049*, 2017.
- [67] D. Knuth. Sorting and searching. *The Art of Computer Programming*, 3:506–549, 1973.
- [68] S. Kumar, T. H. Lai, and A. Arora. Barrier coverage with wireless sensors. *Wireless Networks*, 6(13):817–834, 2007.
- [69] C. Kuratowski. Sur le problème des courbes gauches en topologie. *Fundamenta mathematicae*, 15(1):271–283, 1930.
- [70] D.-T. Lee and F. P. Preparata. Euclidean shortest paths in the presence of rectilinear barriers. *Networks*, 14(3):393–410, 1984.
- [71] Y. Liu, Z. Liu, J. Shi, G. Wu, and C. Chen. Optimization of base location and patrol routes for unmanned aerial vehicles in border intelligence, surveillance, and reconnaissance. *Journal of Advanced Transportation*, 2019, 2019.
- [72] Y. Liu, A. Morgana, and B. Simeone. A linear algorithm for 2-bend embeddings of planar graphs in the two-dimensional grid. *Discrete Applied Mathematics*, 81(1-3):69–91, 1998.

- [73] V. Luca, N. Marín, F. Oliveira, A. Ramírez-Vigueras, O. Solé-Pi, J. Szwarcfiter, and J. Urrutia. Grid straight-line embeddings of trees with a minimum number of bends per path. *Information Processing Letters*, 174:106210, 2022.
- [74] J. N. Marín Nevárez. *Cobertura con balizas en polígonos y poliedros ortogonales*. Tesis de maestría, Universidad Nacional Autónoma de México, 2017.
- [75] N. Megiddo. Linear-time algorithms for linear programming in \mathbb{R}^3 and related problems. *SIAM journal on computing*, 12(4):759–776, 1983.
- [76] A. A. Melkman. On-line construction of the convex hull of a simple polyline. *Information Processing Letters*, 25(1):11–12, 1987.
- [77] T. Michael. Guards, galleries, fortresses, and the octoplex. *The College Mathematics Journal*, 42(3):191–200, 2011.
- [78] A. Morgana, C. P. de Mello, and G. Sontacchi. An algorithm for 1-bend embeddings of plane graphs in the two-dimensional grid. *Discrete applied mathematics*, 141(1-3):225–241, 2004.
- [79] S. Ntafos and L. Gewali. External watchman routes. *The visual computer*, 10(8):474–483, 1994.
- [80] J. O’Rourke. *Art gallery theorems and algorithms*. Oxford University Press Oxford, 1987.
- [81] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, 1985.
- [82] R. C. Prim. Shortest connection networks and some generalizations. *The Bell System Technical Journal*, 36(6):1389–1401, 1957.
- [83] G. Rote. The largest contained quadrilateral and the smallest enclosing parallelogram of a convex polygon. *arXiv preprint arXiv:1905.11203*, 2019.
- [84] E. Schönhardt. Über die zerlegung von dreieckspolyedern in tetraeder. *Mathematische Annalen*, 98(1):309–312, 1928.
- [85] M. I. Shamos. Problems in computational geometry. *Manuscripto no publicado*, 1975.
- [86] M. I. Shamos. *Computational geometry*. PhD thesis, Yale University, 1978.
- [87] M. I. Shamos. The early years of computational geometry—a personal. In *Advances in Discrete and Computational Geometry: Proceedings of the 1996*

AMS-IMS-SIAM Joint Summer Research Conference, Discrete and Computational Geometry—Ten Years Later, July 14–18, 1996, Mount Holyoke College, volume 223, page 313. American Univ in Cairo Press, 1999.

- [88] M. I. Shamos and D. Hoey. Closest-point problems. In *16th Annual Symposium on Foundations of Computer Science (sfcs 1975)*, pages 151–162. IEEE, 1975.
- [89] T. C. Shermer. Recent results in art galleries (geometry). *Proceedings of the IEEE*, 80(9):1384–1399, 09 1992.
- [90] Y. Shiloach. Arrangements of planar graphs on the planar lattice. *PhD thesis, Weizmann Institute of Science*, 1976.
- [91] J. J. Sylvester. A question in the geometry of situation. *Quarterly Journal of Pure and Applied Mathematics*, 1(1):79–80, 1857.
- [92] R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM Journal on Computing*, 16(3):421–444, 1987.
- [93] R. Tamassia and I. G. Tollis. Planar grid embedding in linear time. *IEEE Transactions on circuits and systems*, 36(9):1230–1234, 1989.
- [94] X. Tan. Fast computation of shortest watchman routes in simple polygons. *Information Processing Letters*, 77(1):27–33, 2001.
- [95] X. Tan and T. Hirata. Constructing shortest watchman routes by divide-and-conquer. In *International Symposium on Algorithms and Computation*, pages 68–77. Springer, 1993.
- [96] X. Tan, T. Hirata, and Y. Inagaki. Corrigendum to “An incremental algorithm for constructing shortest watchman routes”. *International Journal of Computational Geometry & Applications*, 9(03):319–323, 1999.
- [97] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160, 1972.
- [98] A. P. Tomás. Guarding thin orthogonal polygons is hard. In *International Symposium on Fundamentals of Computation Theory*, pages 305–316. Springer, 2013.
- [99] G. Toussaint. What is computational geometry. In *Proceedings IEEE, September, 1992*, volume 80, pages 1347–1363. Addison-Wesley, 1992.
- [100] J. Urrutia. Art gallery and illumination problems. In *Handbook of computational geometry*, pages 973–1027. Elsevier, 2000.

- [101] L. G. Valiant. Universality considerations in VLSI circuits. *IEEE Transactions on Computers*, 100(2):135–140, 1981.
- [102] I. van der Hoog, V. Keikha, M. Löffler, A. Mohades, and J. Urhausen. Maximum-area triangle in a convex polygon, revisited. *Information Processing Letters*, page 105943, 2020.
- [103] G. Viglietta. Face-guarding polyhedra. *Computational Geometry*, 47(8):833–846, 2014.
- [104] G. Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les paralléloèdres primitifs. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, 1908(134):198–287, 1908.
- [105] E. Yakıcı. Solving location and routing problem for UAVs. *Computers & Industrial Engineering*, 102:294–301, 2016.
- [106] L. Yanpei, P. Marchioro, and R. Petreschi. At most single-bend embeddings of cubic graphs. *Applied Mathematics*, 9(2):127–142, 1994.