



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

UNIDAD ZACATENCO

DEPARTAMENTO DE COMPUTACIÓN

Tamaño Máximo de un *Thrackle* de Triángulos

TESIS QUE PRESENTA

Santiago León Ortiz

PARA OBTENER EL GRADO DE

Maestro en Ciencias en Computación

DIRECTORA DE LA TESIS

Dra. María Dolores Lara Cuevas

CIUDAD DE MÉXICO

OCTUBRE, 2017

Resumen

La conjetura de Erdős-Faber-Lovász (EFL) lleva más de 40 años abierta, durante este tiempo ha sido de interés en varias áreas de la investigación científica como lo son la teoría de gráficas, la teoría de hipergráficas y la teoría de diseños. Estudiamos una variante de la conjetura EFL, para gráficas geométricas. El caso particular que trabajamos se traduce al problema de encontrar el número máximo de triángulos sobre n puntos, que son disjuntos en aristas y se intersectan dos a dos, a estos conjuntos de triángulos les llamamos *thrackles de triángulos*. Obtenemos el número máximo de triángulos que puede tener un thrackle de triángulos para valores pequeños de n de forma analítica, y utilizamos búsquedas computacionales para valores de n más grandes. Entre los resultados que obtuvimos se encuentran cotas superiores tanto para el tamaño de un thrackle de triángulos, como para el número de thrackles distintos sobre un conjunto de puntos. A partir de estos resultados obtuvimos una cota de la complejidad computacional de hacer una búsqueda exhaustiva de todos los thrackles de triángulos. Los resultados que presentamos en esta tesis, dan evidencia de que el tamaño máximo de un thrackle de triángulos, es el conjeturado en el artículo que propuso el problema originalmente.

Abstract

The Erdős-Faber-Lovász (EFL) conjecture has been open for more than 40 years. During this time the conjecture has been of interest in several areas of mathematics such as graph theory, hypergraph theory, and design theory. In this thesis we study a variant of the conjecture for geometric graphs. The specific case we study can be translated to the problem of finding the maximum number of triangles over a set of n points, that are edge disjoint and intersect each other. We call this kind of triangle sets *triangle thrackles*. We prove analytically the exact size of the maximum size of a triangle thrackle for small values of n , and use computational search to study how triangle thrackles behave for bigger values of n . We present bounds both for the size of a triangle thrackle, and for the number of distinct triangle thrackles over a point set. Based on these results, we obtain a lower bound on the computational complexity of doing an exhaustive search of triangle thrackles. The results presented in this thesis add evidence in support of a conjecture on the maximum size of a triangle thrackle, made in the publication that originally proposed the problem.

Agradecimientos

Agradezco a mi asesora, la Dra. Dolores Lara, por todas las horas invertidas en la revisión y discusión de temas relacionados a la tesis. Además de su excelente guía para no desviar mi atención del problema original. Sus enseñanzas teóricas y de redacción fueron claves en el desarrollo de esta tesis.

Doy gracias a mi familia y amigos, cuya paciente compañía y constantes discusiones, ayudaron a mantenerme motivado y explorar nuevas formas de explicar los problemas.

Finalmente, agradezco al CONACyT y al CINVESTAV por la oportunidad de cursar la maestría, y por los apoyos económicos recibidos durante estos dos años.

Índice general

1. Introducción	9
2. Antecedentes	11
2.1. Gráficas	11
2.2. Gráficas geométricas	13
2.3. Tipos de orden	15
2.4. Thrackles	16
2.5. Árboles	18
3. Estado del arte	21
4. Puntos en posición general	27
4.1. Conjuntos de puntos en posición general	27
4.2. Número de thrackles distintos	36
5. Puntos en posición convexa	39
5.1. Cota inferior del tamaño de un thrackle	39
5.2. Número de thrackles distintos	45
5.3. Tamaño de triángulos	56
6. Conclusiones	61

Índice de figuras

2.1.	Dos representaciones geométricas distintas de la gráfica bipartita completa $K_{2,3}$.	14
2.2.	Ejemplos de polígonos: a) convexo o 1-convexo, b) 2-convexo y c) 3-convexo.	15
2.3.	Tipos de orden para $n=4$, con los valores correspondientes de cada tripleta ordenada.	17
2.4.	a) Dos segmentos que se cruzan. b) Tres segmentos que se intersectan dos a dos.	18
2.5.	Ejemplo de un thrackle con $k = 3$, para un conjunto S de 12 puntos.	18
4.1.	Conjuntos de 8 puntos con mayor número de thrackles de triángulos. Cada uno tiene 120 thrackles.	37
4.2.	Conjuntos de 9 puntos con mayor número de thrackles de triángulos. Cada uno tiene 3156 thrackles.	38
5.1.	Caso $3 n$.	42
5.2.	Caso $3 n - 1$.	43
5.3.	$\frac{n-5}{3}$ triángulos adicionales para $3 n - 2$ con $ A $ impar (no se incluyen los triángulos entre los conjuntos B y C).	44
5.4.	Caso $3 n - 2$ con $ A $ par (no se incluyen los triángulos entre los conjuntos B y C).	45
5.5.	2-factores de la gráfica bipartita completa $K_{3,3}$, con multiconjunto característico $\{3\}$. Esto ilustra que $e_3 = 6$.	47
5.6.	Conteo de los 2-factores de una gráfica bipartita completa $K_{n,n}$ para $n = 7$, con multiconjunto característico $A = \{2, 2, 3\}$.	51
5.7.	Partición del conjunto \mathcal{M} , para $n = 4$.	52
5.8.	Primer subárbol de \mathcal{T} para $n = 4$ y el complemento de uno de los caminos C' con su multiconjunto característico.	53

Índice de tablas

4.1. Número de thrackles de triángulos para conjuntos n puntos con $n \leq 8$	37
5.1. Número de caminos de \mathcal{T} de longitud $n-2$ con complemento A (número de 1-factorizaciones de $K_{n,n}$) y número de 2-factores de $K_{n,n}$ con multiconjunto característico A , para cada $A \in P'(n)$	56
5.2. Número de thrackles maximales en el conjunto de n puntos en posición convexa, clasificados por tamaño de sus triángulos.	59

Índice de algoritmos

1.	Busca un thrackle.	31
2.	Recorre completamente el árbol de búsqueda.	34
3.	Evalúa $L(A)$	50
4.	Calcula el tamaño de un triángulo definido sobre un conjunto S de n puntos en posición convexa etiquetados en sentido horario.	57

Capítulo 1

Introducción

Los conjuntos de puntos en el plano han sido estudiados por los matemáticos desde hace mucho tiempo, en áreas como la combinatoria, la geometría y la teoría de gráficas. En especial, se han estudiado propiedades de estructuras dibujadas sobre conjuntos de puntos. Por ejemplo en 1946, Paul Erdős encontró que dado un conjunto de n puntos, un conjunto de segmentos con extremos en él tal que cada par de segmentos se intersecta, sólo puede tener máximo n segmentos [1].

Esta noción de conjuntos de objetos que se intersectan dos a dos se sigue estudiando porque se relaciona con problemas de coloraciones de gráficas. Se han estudiado diferentes variantes del problema, cambiando los elementos del conjunto por caminos, ciclos hamiltonianos, triángulos disjuntos en vértices o curvas [2, 3]. También se ha cambiado la noción de intersección, siendo las principales las *crossing families* [4] y los *thrackles* [3].

En 1972 Paul Erdős, Vance Faber y László Lovász plantearon la conjetura Erdős-Faber-Lovász (EFL) [5], en el contexto de coloraciones de elementos de conjuntos. El planteamiento de esta conjetura ha sido transformado a conjeturas equivalentes en términos de otras áreas de investigación, como coloraciones de hipergráficas, coloraciones de descomposiciones de la gráfica completa, ordenes, entre otros. En [6] los autores estudian una variante de la conjetura EFL para gráficas geométricas. Esta conjetura propone un valor para el índice cromático de descomposiciones de la gráfica geométrica completa en subgráficas inducidas. El índice cromático se obtiene de asignar a las subgráficas de una descomposición, el menor número de colores de forma que dos subgráficas que se intersectan tengan colores distintos, a esto le llamamos una coloración. Como una coloración de un conjunto de subgráficas que se intersectan dos a dos debe tener asociado a cada una de

las subgráficas un color distinto, el tamaño máximo de dicho conjunto, es una cota inferior del valor del índice cromático de la descomposición. Es decir que el tamaño máximo de un conjunto subgráficas que se intersectan dos a dos, es una cota inferior de la variante de la conjetura EFL para gráficas geométricas.

El problema que estudiamos en esta tesis es un caso particular de esta conjetura planteada en [6]. En nuestro caso, nos limitamos a descomposiciones en triángulos y segmentos. En este caso el tamaño máximo de un conjunto de triángulos que se intersecta dos a dos da una cota inferior del índice cromático de una descomposición en un caso particular. En esta tesis estudiamos el tamaño máximo de estos conjuntos de triángulos que se intersectan dos a dos a los que llamamos *thrackles de triángulos*.

Estudiamos el problema del tamaño máximo de un thrackle de triángulos por medio de búsquedas computacionales. Utilizamos algoritmos de backtracking para obtener valores exactos del número máximo de triángulos en un thrackle sobre conjuntos de puntos de cardinalidad pequeña. Hacemos conteos combinatorios del número de thrackles de triángulos distintos para cada conjunto de puntos obteniendo cotas que se verifican con los resultados experimentales. Además proponemos una caracterización de los conjuntos de puntos con mayor número de thrackles distintos. Estos resultados obtenidos con respecto al número de thrackles, dan cotas inferiores del tiempo de ejecución de búsquedas exhaustivas y muestran la complejidad del problema, para valores de n más grandes.

El contenido de este documento se divide en 5 capítulos. Los siguientes dos capítulos son una introducción al problema que estudiamos, en el capítulo 1 introducimos las definiciones que utilizaremos en el resto de del documento, en el capítulo 2 presentamos un breve resumen de los cambios que ha sufrido la conjetura EFL y presentamos resultados relevantes que ya existen en la literatura. En los siguientes dos capítulos presentamos los resultados obtenidos, primero acerca de conjuntos de puntos en posición general en el capítulo 3 y después acerca de conjuntos de puntos en posición convexa en el capítulo 4. Finalmente en el capítulo 5 resumimos los resultados y trabajo futuro.

Capítulo 2

Antecedentes

En este capítulo presentamos algunos de los conceptos, y la notación, que se utilizarán en la tesis. Hemos tomado todas las definiciones de [7], a menos que se indique lo contrario.

2.1. Gráficas

Una *gráfica* $G = (V, E)$ es una pareja de conjuntos finitos V y E . Los elementos de V son llamados *vértices*, y los elementos de E son parejas (no ordenadas) de vértices llamadas *aristas*. Cada elemento $e = \{u, v\} \in E$ se denota como uv o vu . Decimos que los vértices u y v son *adyacentes*, y que la arista e *incide* en los vértices u y v . Dos aristas distintas son *adyacentes* si tienen algún vértice en común. Generalmente para referirnos a los conjuntos de vértices y de aristas de una gráfica G escribimos $V(G)$ y $E(G)$, respectivamente. Decimos que una gráfica es *no trivial* si tiene dos o más vértices, y es *vacía* si no tiene ninguna arista. El *grado* de un vértice v es el número de aristas en G que inciden en v , y se denota como $\deg(v)$. Le llamamos *k-regular* a una gráfica en la que todos sus vértices tienen grado k .

Muchas veces es de nuestro interés trabajar con recorridos sobre alguna gráfica, por lo que es de utilidad definir una *caminata*. Sea G una gráfica, una *caminata* W de G es una secuencia de vértices de G en la que cada pareja de vértices consecutivos de W son adyacentes en G . Si existe una caminata entre cada par de vértices de G decimos que G es *conexa*. Sea $C = (v_1, \dots, v_n)$ una caminata de G , decimos que C es un *ciclo* si para $i, j \in \{1, \dots, n-1\}$ con $i \neq j$, sucede que $v_i \neq v_j$ y además $v_1 = v_n$. A un ciclo con n vértices lo denotamos como C_n ; a C_3 le llamamos también *triángulo*. Diremos que un

triángulo T está definido sobre G si T es subgráfica de G .

Algunas gráficas son de interés particular y tienen nombres concretos. Una gráfica H es *subgráfica* de G si $V(H) \subseteq V(G)$ y $E(H) \subseteq E(G)$, si $V(H) = V(G)$ entonces H es una subgráfica *generadora* de G . Al conjunto de aristas de una subgráfica generadora 1-regular le llamamos *emparejamiento perfecto*. Una gráfica G es *completa* si toda pareja de vértices distintos de G son adyacentes. Denotamos una gráfica completa con n vértices como K_n . Una gráfica no trivial G es *bipartita* con *bipartición* U, W si su conjunto de vértices se puede dividir en los subconjuntos U y W , tales que $U \cup W = V(G)$ y $U \cap W = \emptyset$, y se cumple además que toda arista de G tiene un vértice en U y otro en W . Una gráfica bipartita con bipartición U, W es *bipartita completa* si todo vértice de U es adyacente a todo vértice de W . Denotamos a la gráfica bipartita completa con bipartición U, W como $K_{s,t}$, donde $s = |U|$ y $t = |W|$. Más adelante haremos uso del siguiente resultado que caracteriza a las gráficas bipartitas. La demostración puede consultarse en [7].

Teorema 1 ([7]). *Una gráfica no trivial G es bipartita si y sólo si G no contiene ningún ciclo impar.*

Sea G una gráfica, un *factor* de G es una subgráfica generadora de G (no necesariamente conexa). A un factor que además es k -regular se le llama *k -factor*. Le llamamos *k -factorización* de G a una colección \mathcal{F} de k -factores de G tales que cada arista de G pertenece a exactamente un k -factor de \mathcal{F} . Le decimos *k -factorizable* a una gráfica que tiene una k -factorización. En particular, nos interesa la siguiente caracterización de los 2-factores de cualquier gráfica G .

Teorema 2. *Sea G una gráfica, y sea F un 2-factor de G . Cada componente conexa de F es un ciclo.*

Demostración. Sea v_1 un vértice de G , y sea F un 2-factor de G . Como un 2-factor es una subgráfica generadora, v_1 debe pertenecer a F . Al ser F 2-regular, v_1 debe tener exactamente 2 vértices adyacentes, le llamamos v_2 a alguno de ellos. El vértice v_2 debe tener exactamente otro vecino distinto a v_1 en F , le llamamos a este vértice v_3 . Continuamos etiquetando de esta forma todos los vértices de la componente conexa de F a la que pertenece v_1 . Como el número de vértices de la componente conexa de F a la que pertenece v_1 es finito, al terminar la etiquetación en el vértice v_k , los vértices v_k y v_1 tienen sólo un vértice adyacente en la secuencia, pero ambos deben ser de grado dos. Esto implica que deben ser adyacentes (es decir que $v_{k+1} = v_1$). Esta secuencia forma un ciclo con k vértices.

Si descartamos los vértices (v_1, \dots, v_k) de F , la subgráfica restante seguirá siendo 2-regular, por lo que podemos tomar otro vértice y repetir el proceso descrito anteriormente. \square

Los elementos de una factorización de una gráfica G son subgráficas generadoras de G . Si eliminamos la condición de que las subgráficas sean generadoras, pero garantizando que sean no vacías, entonces estamos hablando de una descomposición de G . Formalmente, una *descomposición* de G es una colección $\mathcal{D} = \{G_1, \dots, G_k\}$ de subgráficas de G tal que cada arista de G pertenece a exactamente una subgráfica en \mathcal{D} y además $\deg(v) \geq 1$ para $v \in V(G_i)$.

Las descomposiciones de gráficas algunas veces representan objetos matemáticos que se han estudiado en otras áreas de las matemáticas, este es el caso de los sistemas de tripletas de Steiner. Un *sistema de tripletas de Steiner* [8] de tamaño n , denotado como $STS(n)$, es una colección de subconjuntos \mathcal{D} de un conjunto V con $|V| = n$, tal que cada elemento de \mathcal{D} es un conjunto de tamaño 3 y cada pareja de V pertenece a exactamente un elemento de \mathcal{D} . Si pensamos en V como el conjunto de vértices de la gráfica completa K_n , las parejas de vértices son las aristas de K_n , y un sistema de tripletas de Steiner es entonces una descomposición de K_n en triángulos. En esta descomposición cada arista de K_n pertenece a exactamente un triángulo de $STS(n)$.

La existencia de los sistemas de tripletas de Steiner ha sido caracterizada de a partir del tamaño de $STS(n)$. El número de aristas de K_n es $\binom{n}{2}$, entonces $STS(n)$ tiene $\binom{n}{2}/3 = \frac{1}{6}n(n-1)$ tripletas. Cada vértice de K_n es adyacente a $n-1$ vértices, y un vértice de un ciclo es adyacente a 2 vértices, entonces cada vértice pertenece a $\frac{1}{2}(n-1)$ tripletas. Esto prueba que para $STS(n)$, $\frac{1}{6}n(n-1)$ y $\frac{1}{2}(n-1)$ deben ser enteros, o dicho de otro modo que $n \bmod 6 = 1$ o $n \bmod 6 = 3$. El regreso de este resultado fue probado en 1847 por Thomas Kirkman [8]. Todo esto completa la demostración del siguiente teorema.

Teorema 3 (T.P. Kirkman, 1847). *Un sistema de tripletas de Steiner $STS(n)$ existe sí y sólo si $n \bmod 6 = 1$ o $n \bmod 6 = 3$.*

2.2. Gráficas geométricas

Una gráfica puede tener diferentes tipos de representaciones, en particular puede representarse como un dibujo en el plano Euclidiano. En este tipo de representaciones las gráficas heredan propiedades de los objetos del plano

Euclidiano, como por ejemplo la intersección de segmentos o las distancias entre puntos. La teoría de gráficas geométricas estudia estas representaciones de gráficas y sus propiedades.

Definición 1 (Dibujo). Un *dibujo* de una gráfica G es una representación de G en el plano Euclidiano, en la que a cada vértice de $V(G)$ le corresponde un punto distinto de \mathbb{R}^2 , y dos puntos se conectan con una curva de Jordan (una curva que no se interseca a sí misma) si y sólo si sus vértices correspondientes son adyacentes en G . Además, ninguna arista pasa por otro vértice más que por sus vértices extremos.

Un dibujo de G en el que sus aristas están representadas por segmentos de recta es una *gráfica geométrica* G [9], se le conoce también como *representación geométrica* de G . Llamamos a G la gráfica asociada a G . Decimos que G está *dibujada sobre* un conjunto de puntos S si $S = V(G)$. Para evitar casos degenerados y simplificar las explicaciones, suponemos que no hay 3 puntos que sean colineales en el conjunto de vértices de una gráfica geométrica.

De la misma forma en que escribimos K_n como la gráfica completa de n vértices, denotamos como K_n a una representación en el plano de K_n como una gráfica geométrica. Igualmente, con $K_{s,t}$ y C_n nos referimos a representaciones geométricas de gráficas bipartitas completas y ciclos respectivamente.

Decimos que dos gráficas geométricas son *equivalentes* si sus conjuntos de vértices y sus conjuntos de aristas son iguales. Como en el plano Euclidiano sólo existe un segmento de recta entre dos puntos, para que dos gráficas geométricas sean equivalentes, basta con que sus gráficas asociadas sean iguales, y dibujarlas sobre el mismo conjunto de puntos. De esta definición de equivalencia podemos observar que hay tantas representaciones geométricas de una gráfica G , como conjuntos de puntos en el plano. Como un sólo vértice tiene infinitas posibilidades de ser dibujado en el plano, la gráfica G tiene igualmente infinitas representaciones geométricas. La Figura 2.1 ilustra dos representaciones geométricas distintas.



Figura 2.1: Dos representaciones geométricas distintas de la gráfica bipartita completa $K_{2,3}$.

Aunque una gráfica tiene infinitas representaciones geométricas, el es-

tudio de las propiedades de sus representaciones se organiza de acuerdo a las propiedades del conjunto de puntos de sus representaciones. Una elección común para este conjunto son los puntos en posición convexa, a continuación definimos dicho conjunto. Un polígono P es la región del plano delimitada por un número finito de segmentos de recta. Si la intersección de P con cualquier recta es siempre un segmento o vacía, decimos que P es *convexo*. Un conjunto S de puntos en el plano está en *posición convexa* si existe un polígono convexo que tiene a S como su conjunto de vértices. Si el conjunto de vértices de una representación geométrica de una gráfica está en posición convexa, entonces llamamos a dicha gráfica *gráfica geométrica convexa* [9]. Recientemente en [10] se introdujo una generalización de convexidad. Un polígono P es *k -convexo* si la intersección de cualquier recta con P es a lo sumo k segmentos de recta. Decimos que un conjunto S de puntos en el plano es *k -convexo* si existe un polígono k -convexo que tiene a S como conjunto de vértices.

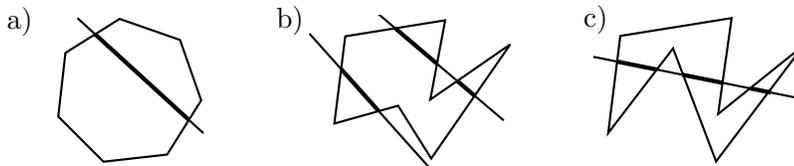


Figura 2.2: Ejemplos de polígonos: a) convexo o 1-convexo, b) 2-convexo y c) 3-convexo.

La Figura 2.2 muestra ejemplos de polígonos k -convexos para k con valores 1, 2 y 3. Notemos que pese a que al polígono de la Figura 2.2b lo intersectan rectas formando solo un segmento de recta, este es 2-convexo porque cualquier recta tiene como intersección con el polígono, a lo sumo 2 segmentos de recta. Por la misma razón el polígono de la Figura 2.2c no puede ser 2-convexo (existe al menos una recta que tiene a tres segmentos como intersección con el polígono). Observamos también que un polígono k -convexo será también $(k + 1)$ -convexo.

2.3. Tipos de orden

Si consideramos secuencias de números reales de tamaño n , pese a que existen infinitas secuencias distintas para un mismo tamaño n , sólo existen dos formas de ordenarlos: de forma ascendente o de forma descendente. Por lo general consideramos que estas dos formas son similares al ser una el reflejo de la otra, así que intuitivamente podemos decir que dichas secuencias

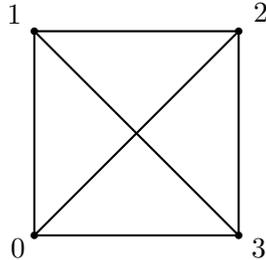
tienen el mismo “*tipo de orden*”. El problema de ordenar una secuencia de números reales es equivalente al de encontrar cuál de las $n!$ permutaciones lleva la secuencia a su forma ordenada. Goodman y Pollack [11] estudiaron generalizaciones de esta noción de orden para puntos que están en cualquier dimensión. Descubrieron que en dimensiones superiores a uno existen diferentes tipos de orden, pero cada conjunto de puntos pertenece sólo a uno de ellos. Si a cada tipo de orden se le asigna una etiquetación canónica, el problema de ordenar un conjunto de puntos en más de una dimensión es similar al de ordenar una secuencia de números reales: ahora se desea buscar una de las $n!$ permutaciones que lleva a las etiquetas de los puntos del conjunto, a la etiquetación canónica de su tipo de orden correspondiente.

Formalmente el *tipo de orden* [12] de un conjunto S de puntos en el plano se define como una función que asocia a cada tripleta ordenada de puntos en S un valor en $\{-1, 1\}$ que representa su orientación. Cuando la tripleta está en sentido antihorario se le asigna 1, y cuando está en sentido horario se le asigna -1. Esto significa que a la tripleta (a, b, c) le corresponde un 1 si c está a la izquierda de la recta dirigida que pasa por ab , y un -1, si por el contrario, c está a la derecha de la recta dirigida que pasa por ab , como ejemplo en la Figura 2.3 mostramos los dos tipos de orden de conjuntos de 4 puntos, y los valores de cada una de las tripletas ordenadas. Decimos que dos conjuntos de puntos S_1 y S_2 son *equivalentes* si existe una biyección de S_1 a S_2 tal que cada tripleta ordenada de S_1 corresponde en orientación con las tripletas ordenadas de S_2 .

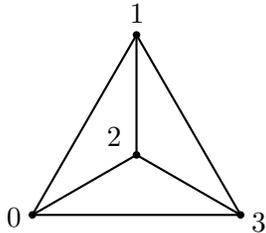
Trabajar con los tipos de orden tiene la ventaja de que reduce el número de gráficas geométricas que hay que estudiar para explorar todas las representaciones geométricas de una gráfica; sólo es necesario explorar un número finito de las mismas. Esto sucede porque los conjuntos de puntos que pertenecen a un mismo tipo de orden por lo general preservan sus propiedades combinatorias, y el problema que estaremos estudiando se basa en este tipo de propiedades. Sin embargo, el número de tipos de orden para un conjunto de n puntos en el plano es al menos $\Omega(2^{n^2/8})$ [11], es decir que el número de tipos de orden crece de forma exponencial con respecto a n .

2.4. Thrackles

Cuando se trabaja con gráficas geométricas, las propiedades del plano dan lugar a la noción de intersección de aristas. De esta noción se derivan problemas que son de interés en el área de la teoría de gráficas geométricas, de los cuales algunos siguen abiertos hoy en día. Uno de los problemas de



012 =	-1	013 =	-1	023 =	-1	123 =	-1
021 =	1	031 =	1	032 =	1	132 =	1
102 =	1	103 =	1	203 =	1	213 =	1
120 =	-1	130 =	-1	230 =	-1	231 =	-1
201 =	-1	301 =	-1	302 =	-1	312 =	-1
210 =	1	310 =	1	320 =	1	321 =	1



012 =	-1	013 =	-1	023 =	-1	123 =	1
021 =	1	031 =	1	032 =	1	132 =	-1
102 =	1	103 =	1	203 =	1	213 =	-1
120 =	-1	130 =	-1	230 =	-1	231 =	1
201 =	-1	301 =	-1	302 =	-1	312 =	1
210 =	1	310 =	1	320 =	1	321 =	-1

Figura 2.3: Tipos de orden para $n=4$, con los valores correspondientes de cada tripleta ordenada.

mayor interés es el estudio de conjuntos de subgráficas de K_n que se intersectan dos a dos. A continuación definimos la intersección de segmentos y de triángulos.

Sean e_1 y e_2 dos aristas de una gráfica geométrica G , decimos que e_1 y e_2 se *crucan* si existe un punto $x \notin V(G)$ tal que $x \in e_1$ y $x \in e_2$. Decimos que e_1 y e_2 se *intersectan*, si se cruzan o si son adyacentes en G . Un *thrackle de segmentos* de G es un conjunto de aristas de G que se intersectan dos a dos. Pese a que estas definiciones están en función de una gráfica geométrica, nuestro objeto de estudio son conjuntos de puntos en el plano. Nótese que cualquier conjunto de puntos naturalmente induce una gráfica geométrica completa, por lo que siempre consideramos las definiciones en términos de gráficas completas.

Sea K_n una gráfica geométrica completa dibujada sobre un conjunto S de n puntos en el plano. Sea T un conjunto de k triángulos definidos sobre K_n . Decimos que los triángulos se *intersectan* si tienen un vértice en común o dos de sus aristas se cruzan.

Definición 2 (Thrackle de triángulos). Sea T un conjunto de k triángulos definidos sobre la gráfica geométrica completa K_n dibujada sobre el conjunto

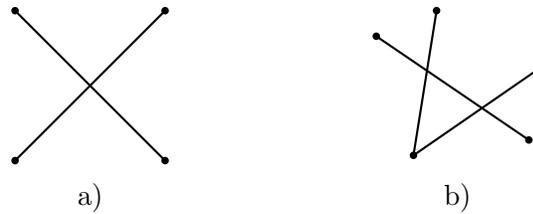


Figura 2.4: a) Dos segmentos que se cruzan.
b) Tres segmentos que se intersectan dos a dos.

de puntos S . Decimos que T es un *thrackle de triángulos* del conjunto de puntos S , de *tamaño* k , si cumple con las siguientes propiedades:

1. Los triángulos en T no tienen aristas en común.
2. Cada par de triángulos en T se intersectan.

Sea T un thrackle de triángulos del conjunto S , si el tamaño de T es el máximo posible, entonces decimos que T es un *thrackle maximal* (un thrackle maximal no necesariamente es único). Definimos el número T_n como el tamaño máximo que tienen los thrackles maximales de cada conjunto de n puntos en el plano. En este trabajo de tesis nos interesa determinar el valor T_n .

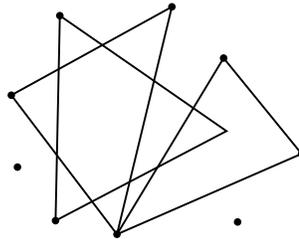


Figura 2.5: Ejemplo de un thrackle con $k = 3$, para un conjunto S de 12 puntos.

2.5. Árboles

El concepto de *árbol* existe en computación como una estructura de datos en la cual se almacena información. En teoría de gráficas un *árbol enraizado* es una gráfica conexa que no contiene ciclos y que tiene un vértice distinguido al que se le llama raíz. En computación, generalmente queremos trabajar

con árboles enraizados, en los que cada vértice es un elemento de información. Como los vértices de una gráfica son distintos entre sí, pero es común querer representar información con repeticiones, preferimos definir un árbol en términos de *nodos*.

Definimos un *árbol* con nodos en el conjunto H como una secuencia finita \mathcal{T} , que contiene elementos a los que llamamos *nodos*. La secuencia \mathcal{T} contiene un nodo distinguido al que llamamos *raíz* de \mathcal{T} . Decimos que una *referencia* a un nodo n es un entero que representa la posición de n en la secuencia \mathcal{T} . Cada *nodo* n consiste de un elemento de H al que llamamos *valor* o *contenido* de n y una secuencia ordenada de referencias a nodos distintos (distintos tanto a n como entre sí), a estos nodos les llamamos *hijos* de n . Ningún nodo en \mathcal{T} contiene al nodo raíz como hijo, y cada nodo (excepto la raíz) es hijo de exactamente un nodo. Al nodo que contiene una referencia a n le llamamos *padre* de n (la raíz es el único nodo sin padre). A los nodos que no tienen hijos les llamamos *hojas*.

Como la secuencia de referencias a los hijos de cada nodo debe ser única, un árbol \mathcal{T} puede verse como un conjunto de nodos distintos entre sí. Si formamos la gráfica G que tiene como conjunto de vértices al conjunto de nodos de \mathcal{T} , distinguiendo la raíz de \mathcal{T} como raíz de G , y existe una arista en G entre dos nodos si y sólo si uno es hijo del otro. Entonces G es un árbol enraizado puesto que si hubiese un ciclo, entonces algún nodo sería hijo de otros dos o la raíz tendría padre, y ambas situaciones son imposibles por la definición de árbol. Esto muestra la relación entre la definición que utilizamos de árbol, y la de árbol enraizado en teoría de gráficas. Utilizando la terminología introducida para gráficas, decimos que una caminata en G que inicia en la raíz, es un *camino* de \mathcal{T} . Por la definición de árbol, cada nodo n tiene exactamente un camino que tiene a n como nodo final, usualmente nos referimos a este camino como el camino del nodo n .

La longitud de un camino de \mathcal{T} es el número de aristas entre los nodos que lo componen, es decir que un camino de longitud c tiene $c + 1$ nodos. Definimos la *altura* de \mathcal{T} como la mayor longitud de un camino de \mathcal{T} . Sea C un camino de \mathcal{T} que contiene al nodo n , le llamamos *ancestros* de n a los nodos que preceden a n en C . Decimos que dos nodos de \mathcal{T} están en el mismo *nivel* si sus caminos tienen la misma longitud.

Capítulo 3

Estado del arte

El problema planteado en esta tesis proviene de una variación de la conjetura de Erdős-Faber-Lovász (EFL) para gráficas geométricas. La conjetura original EFL fue planteada en 1972 y sigue sin ser resuelta, la forma en que se propuso originalmente es la siguiente.

Conjetura 1 (EFL[6]). *Sean n conjuntos A_i con $|A_i| = n$ y $1 \leq i \leq n$, si $|A_i \cap A_j| \leq 1$ para $1 \leq i < j \leq n$, entonces los elementos de la unión $\bigcup_{i=1}^n A_i$ se pueden colorear con n colores de forma que cada conjunto tenga elementos de todos los colores.*

Este problema se puede traducir a un problema de coloraciones de hipergráficas. A continuación introducimos las definiciones de hipergráficas y coloraciones de hipergráficas, tomadas de [13]. Una *hipergráfica* $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ es un conjunto no vacío \mathcal{V} cuyos elementos son llamados vértices, junto con un conjunto \mathcal{E} de subconjuntos no vacíos de \mathcal{V} llamados *hiperaristas*. Cada vértice de \mathcal{V} pertenece a al menos a una hiperarista. Una hipergráfica es *lineal* si la intersección de cualquier par de hiperaristas es un vértice o es vacía. Una *k -coloración de los vértices* de una hipergráfica \mathcal{H} es una función que asocia a cada vértice de \mathcal{H} un color del conjunto $\{1, \dots, k\}$, de forma que los vértices de cada hiperarista tengan colores distintos. El *número cromático* $\chi(\mathcal{H})$ de \mathcal{H} es el entero más pequeño k para el cual existe una k -coloración de los vértices de \mathcal{H} . Una *k -coloración de las hiperaristas* de una hipergráfica \mathcal{H} es una función que asocia a cada hiperarista de \mathcal{H} un color del conjunto $\{1, \dots, k\}$ tal que si dos hiperaristas tienen un vértice en común, entonces tienen colores diferentes. El *índice cromático* $\chi'(\mathcal{H})$ de \mathcal{H} es el entero más pequeño k para el cual existe una k -coloración de las hiperaristas de \mathcal{H} . Utilizando estas definiciones podemos escribir la conjetura EFL de la siguiente forma:

Conjetura 2 ([13]). *Si \mathcal{H} es una hipergráfica lineal con n hiperaristas, cada hiperarista con n vértices, entonces*

$$\chi(\mathcal{H}) = n.$$

Llamamos a una hipergráfica *intersectante* (en inglés *intersecting*) si cualquier pareja de hiperaristas tiene intersección no vacía. Una hipergráfica, con n hiperaristas de tamaño n en la que cada par de hiperaristas tiene intersección de cardinalidad 1, es una *n -cúmula* (en inglés *n -cluster*). En [13] los autores dan una secuencia de hipergráficas que permite transformar una hipergráfica lineal \mathcal{H} en una hipergráfica intersectante \mathcal{H}' . Esta construcción permite extender una n -coloración de vértices de \mathcal{H}' a una n -coloración de vértices de \mathcal{H} . Si \mathcal{H} tiene además n hiperaristas de tamaño n , entonces \mathcal{H}' es una n -cúmula y por lo tanto los números cromáticos de \mathcal{H} y \mathcal{H}' son iguales. Esta construcción prueba que la Conjetura 2 es equivalente a la siguiente.

Conjetura 3 ([13]). *Si una hipergráfica \mathcal{H} es una n -cúmula entonces*

$$\chi(\mathcal{H}) = n.$$

Utilizando esta forma de la conjetura EFL, una búsqueda computacional exhaustiva que intenta verificar la Conjetura 2 ya no necesita encontrar n -coloraciones para todas las posibles hipergráficas lineales, sino que es suficiente con explorar todas las n -cúmulas. Esta fue de hecho la técnica utilizada por los autores de [14] para probar que la conjetura EFL es cierta para $n \leq 12$.

Sea \mathcal{H} una hipergráfica con vértices $\mathcal{V} = \{v_1, \dots, v_n\}$ y m hiperaristas $\mathcal{E} = \{E_1, \dots, E_m\}$. La hipergráfica dual $\mathcal{H}^* = (\mathcal{V}^*, \mathcal{E}^*)$ de \mathcal{H} es la hipergráfica que tiene como vértices $\mathcal{V}^* = \{e_1, \dots, e_m\}$ a las hiperaristas de \mathcal{H} , y que tiene una hiperarista V_i por cada vértice v_i de \mathcal{H} de la siguiente forma $V_i = \{e_j : v_i \in E_j \in \mathcal{E}\}$. Notemos que una n -coloración de los vértices de \mathcal{H} induce una n -coloración de las hiperaristas de \mathcal{H}^* .

Definimos una *k -coloración* de una descomposición de una gráfica G como una función que le asigna a cada elemento de la descomposición un color, tal que que si dos subgráficas de G tienen un vértice en común, entonces tienen colores distintos. El *índice cromático* de una descomposición D de una gráfica es el entero más pequeño k para el cual existe un k -coloración de D . Si tomamos la hipergráfica dual de una n -cúmula obtenemos una hipergráfica que tiene n vértices y que tiene al menos una hiperarista entre cada par de vértices. Por lo tanto podemos representar a la hipergráfica dual de una n -cúmula como una gráfica completa K_n , y pensar en sus hiperaristas como una descomposición de K_n en subgráficas inducidas. Entonces,

una n -coloración de una n -cúmula se convierte en una n -coloración de una descomposición de K_n en subgráficas inducidas. Por lo anterior en [15] los autores llevan la conjetura EFL a la siguiente forma:

Conjetura 4 ([15]). *Toda descomposición de K_n en subgráficas inducidas tiene índice cromático n .*

Los autores de [6] se basan en la forma anterior de la conjetura EFL para proponer un nuevo problema, similar a la conjetura EFL pero sobre gráficas geométricas. En este nuevo problema los autores usan una definición de coloraciones de descomposiciones de la gráfica geométrica completa K_n que toma en cuenta la intersección de aristas en el plano. En esta definición, en una coloración de una descomposición, dos subgráficas que tienen algún par de aristas que se intersecta no puedan tener el mismo color. La existencia de un conjunto de k subgráficas completas de la gráfica geométrica completa K_n que se intersectan dos a dos, implica que la descomposición de K_n en dichas subgráficas debe colorearse con al menos k colores. El estudio del índice cromático para el caso particular de descomposiciones de K_n en segmentos y triángulos (en otras palabras K_2 y K_3 , que son subgráficas inducidas de K_n), se convierte en el problema de encontrar thrackles de triángulos de gran tamaño. Esto porque sabemos que un thrackle de segmentos no puede tener más segmentos que vértices, es decir que no aporta más que un factor lineal al índice cromático de una descomposición de K_n . Recordemos que T_n representa el tamaño del thrackle de triángulos de mayor tamaño, definido sobre cualquier conjunto de n puntos. Es de nuestro interés la siguiente conjetura planteada en [6].

Conjetura 5 ([6]). *Para todo conjunto S de n puntos,*

$$T_n < \frac{n^2}{9} + 2.$$

La Conjetura 5 proviene de uno de los teoremas presentados en [6]. Dicho teorema da una cota inferior del tamaño de un thrackle de triángulos en conjuntos de puntos en posición convexa. La conjetura se basa en la suposición de que la cota inferior del tamaño de un thrackle para puntos en posición convexa, también es una cota superior para cualquier conjunto de puntos salvo por un valor constante. El teorema del que proviene la conjetura es el siguiente.

Teorema 4 ([6]). *Sea n un entero positivo tal que $3|n$. Sea S un conjunto de n puntos en posición convexa. Existe un thrackle de triángulos en S de tamaño $\frac{n^2}{9} + 1$.*

Existen otros artículos con resultados relacionados con el problema de encontrar conjuntos de triángulos que se intersectan, algunos de los cuales dan otras cotas para el tamaño de un thrackle. Por ejemplo en [16] se estudian conjuntos de triángulos vacíos. Un triángulo con vértices en un conjunto de puntos S es *vacío* si no contiene en su interior a ningún punto de S . Uno de los problemas tratados en [16] es el siguiente: dado un conjunto S de n puntos en el plano, encontrar el tamaño máximo δ de un conjunto de triángulos que no tienen artistas en común y que son *vacíos*. Con respecto a este problema los autores presentan una familia de conjuntos de puntos para los que δ es a lo sumo $O(n \log n)$.

Otro problema tratado en [16] es el de encontrar, para un conjunto de puntos S , cuántos triángulos con vértices en S , disjuntos en aristas, contienen a un mismo punto $q \notin S$ en su interior. Para un conjunto de puntos S y un punto en el plano $q \notin S$, denotamos como $\tau(q)$ a la mayor cantidad de triángulos arista disjuntos con vértices en S que contienen a q . En [16] los autores presentan el siguiente resultado.

Teorema 5 ([16]). *Para cualquier conjunto S de n puntos en el plano, existe un punto $q \notin S$ tal que $\frac{n^2}{12} \leq \tau(q) \leq \frac{n^2}{9}$. Y además, para cualquier punto $q \notin S$, $\tau(q) \leq \frac{n^2}{9}$.*

Para puntos en posición convexa, el Teorema 5 implica una cota inferior para el tamaño máximo de un thrackle de triángulos. Considérese un conjunto S de puntos en posición convexa, cualquier conjunto de triángulos con vértices en S y que contienen a un mismo punto en su interior, necesariamente se intersectan. Por lo tanto S tiene un thrackle con al menos $\frac{n^2}{12}$ triángulos.

Este último problema de encontrar el número máximo de triángulos que contienen a un punto en su interior, fue tratado también en [17]. Los autores demostraron que cualquier punto del plano debe estar contenido en al menos $\frac{n^3}{27}$ triángulos no necesariamente disjuntos ni en aristas ni en vértices.

En cuanto a problemas relacionados con encontrar conjuntos de triángulos que se intersecten dos a dos, en [2] se estudia el siguiente problema: encontrar el tamaño de un conjunto de triángulos con extremos en un conjunto de puntos S , tales que se intersectan dos a dos, y no tienen ningún vértice en común. Los autores probaron que el tamaño de dicho conjunto es siempre al menos $\lfloor \frac{n}{6} \rfloor$ para cualquier conjunto de puntos en posición general. Como la cantidad máxima de triángulos que puede haber en uno de estos conjuntos es $\lfloor \frac{n}{3} \rfloor$, su tamaño máximo es de orden $\Theta(n)$. Es decir, que el problema de calcular el tamaño de un conjunto de triángulos en el que los

vértices son disjuntos (a diferencia de un thrackle en el que sólo las aristas de los triángulos son disjuntas) ya está resuelto.

Otros resultado que proviene de la teoría de diseños, da una cota para el tamaño máximo de conjuntos de triángulos disjuntos en aristas. Como todo thrackle maximal es un conjunto de triángulos disjuntos en aristas, el tamaño de un thrackle maximal no podrá ser mayor a la máxima cardinalidad de un conjunto de triángulos disjuntos en aristas. De este resultado presentado en [18] obtenemos el Teorema 6 que es una cota superior para el tamaño de un thrackle de triángulos en un conjunto de n puntos en posición general, que recordemos hemos denotado como T_n en la sección anterior.

Teorema 6 ([19, 18]). *Para cualquier entero n el valor de T_n cumple que:*

$$T_n \leq \begin{cases} \frac{n^2}{6} - \frac{n}{3} & \text{si } 6|n \text{ o } 6|n - 2 \\ \frac{n^2}{6} - \frac{n}{6} & \text{si } 6|n - 1 \text{ o } 6|n - 3 \\ \frac{n^2}{6} - \frac{n}{3} - \frac{1}{3} & \text{si } 6|n - 4 \\ \frac{n^2}{6} - \frac{n}{6} - \frac{4}{3} & \text{si } 6|n - 5 \end{cases}$$

En resumen, de la literatura sabemos que un thrackle de triángulos en un conjunto de n puntos en posición convexa debe tener al menos $\frac{n^2}{12}$ triángulos, si $3|n$ entonces existe un thrackle con $\frac{n^2}{9} + 1$ triángulos. Por otro lado, aunque se cree que un thrackle en un conjunto de n puntos en posición general debe tener a lo sumo $\frac{n^2}{9} + 5$ triángulos, sólo la expresión dada en el Teorema 6 ha sido probada, y es superior a lo esperado.

Capítulo 4

Puntos en posición general

En [12] los autores presentaron una base de datos [20] de todos los conjuntos de puntos correspondientes a cada uno de los tipos de orden para valores pequeños de n . Esta base de datos permite hacer búsquedas exhaustivas, de donde pueden obtenerse contraejemplos, o casos base, para resultados teóricos de problemas combinatorios sobre conjuntos de puntos.

En este trabajo de tesis utilizamos esta base de datos para hacer búsquedas computacionales sobre conjuntos de hasta 10 puntos. Buscamos dar información con respecto al tamaño máximo de thrackles de triángulos, y encontrar resultados combinatorios que puedan ser generalizados.

En este capítulo presentamos los resultados obtenidos acerca del problema de encontrar el tamaño máximo de un thrackle de triángulos sobre un conjunto de n puntos en posición general, para valores pequeños de n .

4.1. Conjuntos de puntos en posición general

Para conjuntos de n puntos con $n \leq 7$, mostramos a continuación los valores del tamaño máximo de un thrackle de triángulos. Retomando la notación introducida anteriormente, al tamaño máximo de un thrackle de triángulos en un conjunto de n puntos en posición general, lo denotamos como T_n . Para conjuntos de 3 y 4 puntos, sólo es posible formar un único triángulo, ya que un thrackle sólo puede tener triángulos que no tengan aristas en común, por lo que $T_n = 1$, para ambos casos. Mostramos a continuación las demostraciones para los demás valores de n .

Teorema 7. *Para $n = 5$, $T_n = 2$*

Demostración. Sea S un conjunto de 5 puntos en el plano. Hay al menos 2 triángulos con vértices en S que no tienen ninguna arista en común. Los

triángulos deben compartir exactamente un vértice pues no hay suficientes puntos para que sean disjuntos. Por lo tanto forman thrackle de triángulos de tamaño 2.

Probamos ahora, por contradicción, que no es posible agregar otro triángulo a este thrackle. Sean t_1 y t_2 los triángulos del thrackle. Supongamos que existe un tercer triángulo t_3 que pertenece al thrackle. Como los 5 puntos del conjunto S ya forman parte de t_1 o de t_2 , los vértices de t_3 deben pertenecer a t_1 o a t_2 . Esto implica que t_3 debe tener 2 vértices en común ya sea con t_1 o con t_2 . Lo que significa que tienen una arista en común con dicho triángulo, pero hemos supuesto que t_3 no comparte aristas ni con t_1 ni con t_2 . Por lo tanto el thrackle es máximo. \square

Teorema 8. *Para $n = 6$, $T_n = 4$*

Demostración. Sea S un conjunto de 6 puntos en el plano. Por el teorema anterior, existe un thrackle $T_2(S)$ con dos triángulos t_1 y t_2 . A partir de este thrackle, a continuación mostramos cómo construir dos triángulos t_3 y t_4 que permiten crear un thrackle $T_4(S)$.

Los triángulos t_1 y t_2 pueden ser de 2 formas: 1) no tener ningún vértice en común e intersectarse únicamente en sus aristas, o 2) tener un vértice en común. En el primer caso, no es posible agregar otro triángulo que sea disjunto en aristas con t_1 o con t_2 . Consideremos ahora el caso 2 y supongamos que t_1 y t_2 tienen un vértice en común. Sea t_3 un triángulo formado por un vértice de t_1 , un vértice de t_2 y el vértice que no pertenece ni a t_1 ni a t_2 . Como cada vértice tiene 5 aristas incidentes en él, entonces no puede pertenecer a más de 2 triángulos. Después de agregar el triángulo t_3 , sólo restan 3 vértices que no pertenecen a dos triángulos, estos vértices forman el cuarto triángulo t_4 . Los triángulos t_1 , t_2 , t_3 y t_4 forman un thrackle de triángulos pues cada triángulo comparte un vértice con los otros tres. Además no es posible agregar más triángulos porque cada vértice ya pertenece a exactamente dos. Por lo tanto, cualquier conjunto de 6 puntos en el plano tiene un thrackle de tamaño 4. Es decir, $T_6 = 4$. \square

Teorema 9. *Para $n = 7$, $T_n = 7$*

Demostración. Como se cumple que para $n = 7$, $n \bmod 6 = 1$ entonces existe un sistema de tripletas de Steiner $STS(7)$, que induce un conjunto de triángulos disjuntos en aristas de tamaño 7. Un sistema de tripletas de Steiner utiliza todas las aristas posibles en el conjunto de puntos por lo que cada vértice pertenece a 3 triángulos. Esto implica que cada triángulo tiene un vértice en común con otros 6 triángulos distintos, 2 triángulos en

cada uno de sus vértices. Como el número total de triángulos es 7, cualquier pareja de triángulos tiene un vértice en común, y por lo tanto se intersectan. Entonces cualquier sistema de tripletas de Steiner de tamaño 7 es un thrackle de triángulos para cualquier conjunto de 7 puntos. \square

Para casos más grandes hicimos búsquedas computacionales sobre la base de datos de tipos de orden [20]. La base de datos contiene un conjunto de n puntos para cada uno de los tipos de orden con $3 \leq n \leq 10$. Todas las pruebas se hicieron sobre una computadora con un procesador Intel Core i7-6700HQ a 2.6GHz, con 16GiB de RAM, utilizando una distribución de Linux basada en Ubuntu 16.04. Aunque el procesador tiene 4 núcleos, todas las pruebas y tiempos de ejecución corresponden a algoritmos sin paralelización.

La técnica utilizada para hacer las búsquedas fue la de backtracking o de vuelta atrás. Un algoritmo de backtracking es una técnica que se utiliza para resolver problemas en los que se desea encontrar una secuencia (x_1, \dots, x_n) , donde los elementos pertenecen a algún dominio $x_i \in D$ de números enteros y para la cual alguna propiedad $P(x_1, \dots, x_n)$ es verdadera [21]. La idea consiste en verificar propiedades parciales $P_l(x_1, \dots, x_l)$ con $1 \leq l < n$ y sólo continuar extendiendo la secuencia con aquellos elementos que harían que la propiedad P_l sea verdadera. Por esto es necesario que las propiedades parciales cumplan que:

1. $P_l(x_1, \dots, x_l)$ es verdadera si $P_{l+1}(x_1, \dots, x_{l+1})$ es verdadera.
2. $P_l(x_1, \dots, x_l)$ es fácil de verificar si $P_{l-1}(x_1, \dots, x_{l-1})$ es verdadera.

Este tipo de búsqueda evita que probemos todas las secuencias de tamaño n : al encontrar un x_l que hace la propiedad P_l falsa, ya no intentamos añadir ningún otro de los elementos x_{l+1}, \dots, x_n . La razón por la que se llama algoritmo de backtracking es porque cuando no hay más elementos que pueden tomar la posición siguiente en la secuencia (x_1, \dots, x_l) , retrocedemos, eliminando a x_l y lo cambiamos por otro elemento de D . Al final, la ejecución del algoritmo puede representarse por medio de un árbol con nodos en $D \cup \{\emptyset\}$, en el que la raíz es el único nodo que tiene como valor a \emptyset . De forma que para nodo x_l , la secuencia de valores (x_1, \dots, x_l) , que representa el camino del nodo x_l , hace verdadera la propiedad parcial $P_l(x_1, \dots, x_l)$. A este árbol asociado a un algoritmo de backtracking le llamamos *árbol de búsqueda*.

El tipo de recorrido de los nodos del árbol de búsqueda que hace un algoritmo de backtracking se conoce como recorrido en profundidad porque desciende hasta encontrar una hoja y entonces comienza a retroceder.

Para aplicar esta técnica al problema de encontrar un thrackle de cierto tamaño k , generamos un arreglo A con los $\binom{n}{3}$ triángulos y le asignamos a cada uno el entero correspondiente a su posición en A . El dominio será $D = \{1, \dots, \binom{n}{3}\}$ y la propiedad $P_l(x_1, \dots, x_k)$ será verdadera si los triángulos $\{A[x_1], \dots, A[x_k]\}$ forman un thrackle. Observamos que las propiedades 1 y 2 se cumplen para esta propiedad parcial P_l . El Algoritmo 1 muestra el pseudocódigo de la implementación del procedimiento que encuentra un thrackle de tamaño k .

El Algoritmo 1 no incluye algunos detalles de implementación para facilitar su comprensión, pero dependiendo de cómo se representen los arreglos B y C (ver pseudocódigo) los tiempos de ejecución pueden variar significativamente. En la implementación utilizada para las pruebas el arreglo B es una lista ligada que sólo contiene los elementos con valor 1. Los arreglos C_i fueron todos implementados en una sola pila C que puede tener a lo sumo $\binom{n}{3}$ elementos. Estas estructuras de datos ayudan a reducir el tiempo de ejecución a medida que aumenta el valor de l durante la ejecución.

Independientemente de los detalles de la implementación, el tiempo de ejecución de un algoritmo de backtracking puede tener comportamientos muy distintos dependiendo de la instancia del problema que se está resolviendo. Sin embargo, en general el tiempo de ejecución está relacionado con el número de nodos que tiene el árbol que representa la ejecución del algoritmo para una instancia del problema. Por ejemplo para el caso $n = 8$ y el conjunto de puntos en posición convexa, el Algoritmo 1 recorre 713,034 nodos del árbol de búsqueda en 0.1 segundos, antes de encontrar el primer thrackle.

Otro factor que afecta el tiempo de ejecución es el orden en que se ubican los triángulos inicialmente en el arreglo A . En la ejecución del algoritmo mencionada en el párrafo anterior ordenamos los triángulos lexicográficamente de acuerdo a las etiquetas de los vértices en la base de datos. Si en cambio, ordenamos el arreglo A aleatoriamente y hacemos 1,000 ejecuciones distintas del Algoritmo 1, en promedio se recorren 120,132.41 nodos antes de encontrar el primer thrackle en el conjunto de 8 puntos en posición convexa, tardando en promedio 10.67 milisegundos en cada ejecución. Esto da evidencias para creer que el orden lexicográfico no es una buena elección para minimizar el tiempo de ejecución, porque en promedio se recorren 6 veces menos nodos que cuando se utiliza el orden lexicográfico.

Por el Teorema 6 sabemos que para $n = 8$, $T_n \leq 8$. Es decir que si existiera un thrackle de tamaño 8 para cada tipo de orden, esto probaría que $T_n = 8$ para $n = 8$. Si suponemos que el Teorema 4 se aproxima al valor del tamaño mínimo de un thrackle para los casos en que $n \nmid 3$ y el

Algoritmo 1: Busca un thrackle.

Input: Conjunto S de n puntos y tamaño k del thrackle deseado.

Output: Arreglo T de k triángulos que forman un thrackle o \emptyset si no existe un thrackle de tamaño k .

```

1 Function Encuentra-Thrackle( $S, k$ )
2    $A \leftarrow$  arreglo con los  $\binom{n}{3}$  triángulos en  $S$ .
3    $B \leftarrow$  arreglo con 1 o 0 de tamaño  $\binom{n}{3}$ .
4    $C_1, \dots, C_l \leftarrow$  Cada  $C_i$  es un conjunto de enteros vacío.
5    $T[0] \leftarrow A[0]$ 
6    $t \leftarrow 0$            /* Índice del último triángulo elegido. */
7    $l \leftarrow 1$          /* Triángulos en el thrackle  $T$ . */
8   while  $l > 0$  do
9     if  $l = k$  then
10      | return  $T$ 
11     else
12       for cada  $i$  tal que  $B[i] = 1$  do
13         | if  $A[i]$  tiene aristas en común con  $A[t]$  then
14           |    $B[i] \leftarrow 0$  y agrega  $i$  a  $C_l$ .
15           |   continue
16         | if  $A[i]$  no interseca a  $A[t]$  then
17           |    $B[i] \leftarrow 0$  y agrega  $i$  a  $C_l$ .
18         | if Existe  $j$  tal que  $B[j] = 1$  then
19           |    $t \leftarrow$  Menor  $j$  tal que  $B[j] = 1$ .
20         | else
21           |    $t \leftarrow$  NULL.
22       Loop
23         | if  $t \neq$  NULL then
24           |    $T[l] \leftarrow A[t]$ 
25           |    $l \leftarrow l + 1$ 
26           |   break
27         | else
28           |    $l \leftarrow l - 1$ 
29           |   if  $l \geq 0$  then
30             |    $t \leftarrow$  Índice del triángulo  $T[l]$  en  $A$ .
31             |    $B[j] \leftarrow 1$  Para cada  $j$  en  $C_l$ .
32             |    $C_l \leftarrow \emptyset$ 
33           |   else
34             |   break
35   return  $\emptyset$ 

```

conjunto de puntos no está en posición convexa, entonces esperamos que haya al menos un thrackle con 8 triángulos sobre cada conjunto de 8 puntos. Por esto, ejecutamos el Algoritmo 1 con $n = 8$ y $k = 8$ para cada uno de los 3315 conjuntos de puntos que representan los de tipos de orden para $n = 8$.

La ejecución del Algoritmo 1, ordenando aleatoriamente el arreglo A para cada tipo de orden, encontró un thrackle de tamaño 8 para cada conjunto de 8 puntos, en un total de 45.78 segundos, recorriendo en promedio 152,462.69 nodos antes de encontrar cada thrackle. Por lo tanto, demostramos computacionalmente el siguiente teorema.

Teorema 10. *Para $n = 8$, $T_n = 8$.*

Aunque el Algoritmo 1 requiere poco tiempo para encontrar un thrackle, esto sólo ocurre si el thrackle que buscamos existe. En caso contrario, por ejemplo cuando utilizamos un valor de k demasiado grande, el tiempo de ejecución crece significativamente. Por ejemplo, para $n = 8$, si intentamos buscar un thrackle de tamaño 9, el tiempo de ejecución es de 54.6 segundos, y se recorren 77,572,239 nodos, para obtener como resultado que no existe tal thrackle. Esto se vuelve un inconveniente si queremos aplicar el mismo método para hallar el valor de T_n para $n = 9$, porque en este caso la cota superior y la cota inferior están más distantes que en el caso $n = 8$. En el caso $n = 8$ sólo fue necesario hacer la búsqueda para $k = 8$, y no tomó mucho tiempo porque el valor esperado resultó ser el correcto y nunca se ejecutó el Algoritmo 1 con una instancia que causara el peor tiempo de ejecución. Y aún así, si hubiésemos adivinado incorrectamente, y no hubiese existido un thrackle de tamaño 8 para conjuntos de 8 puntos (salvo en posición convexa), el tiempo de ejecución esperado era de aproximadamente 2 días, que es aceptable dentro del tiempo disponible.

Por el contrario, para $n = 9$ incluso suponiendo que el Teorema 4 aproxima la cota inferior de un thrackle de triángulos para conjuntos de puntos en posición general, y por el Teorema 6, obtenemos que $9 \leq T_n \leq 12$. Es decir que al menos habría que hacer búsquedas para $k = 10$, $k = 11$ y $k = 12$, con el riesgo de que en algunas de ellas, para todo tipo de orden, no exista un thrackle de tamaño k . Esto causaría que el Algoritmo 1, se llame para todo tipo de orden con una instancia que causa el peor tiempo de ejecución. Si consideramos que para $n = 9$ y el conjunto de puntos en posición convexa, el tiempo de ejecución del Algoritmo 1 es superior a 30 minutos, y que hay 158,817 tipos de orden, obtenemos una estimación de más de 9 años para la búsqueda de un thrackle para cada tipo de orden de tamaño 9.

Cuando ocurre el peor caso del Algoritmo 1 estamos recorriendo todo el árbol de búsqueda. Aún si el algoritmo sólo regresara un conjunto vacío, en el

tiempo de ejecución que tomó para llegar a ese resultado, se podría obtener mucha más información si la estuviésemos almacenando de alguna forma. Al recorrer el árbol por completo podríamos obtener, por ejemplo, el tamaño máximo de un thrackle para ese tipo de orden, así como todos los thrackles de ese tamaño. Es decir que haciendo esta modificación al Algoritmo 1, para calcular el valor de T_n en el caso $n = 9$ bastaría con hacer una sola búsqueda. Esto a diferencia de al menos 3 ejecuciones del Algoritmo 1 para $k = 10$, $k = 11$ y $k = 12$.

Cambiando el final de ejecución de la línea 10 del Algoritmo 1 por el incremento de un entero, podemos contar el número total de secuencias de triángulos encontradas que representan un thrackle. Si ejecutamos esta versión modificada sobre el conjunto de 8 puntos en posición convexa el resultado son 4,838,400 secuencias de 8 triángulos. Sin embargo, una inspección más cercana de los resultados muestra que en realidad en el árbol se exploran todas las permutaciones de cada secuencia de triángulos, de hecho $4,838,400/8! = 120$. Esto significa que sólo hay 120 thrackles distintos.

Tomando en cuenta esta observación, podemos mejorar el algoritmo para que recorra menos nodos haciendo que sólo se explore una de las permutaciones de cada thrackle. Una posible elección de dicha permutación, es la permutación que consiste de triángulos con índices ascendentes en el arreglo A . El Algoritmo 2 recorre todo el árbol de búsqueda para un n dado y un conjunto de puntos S , y regresa la altura del árbol, es decir T_n .

Las principales diferencias entre los Algoritmos 1 y 2 son condiciones relacionadas con el valor de t . De ellas, la única necesaria para reducir el número de nodos explorados son las de las líneas 18 y 19 del Algoritmo 2. Sin embargo, es importante también cambiar las líneas 10 y 32 del Algoritmo 2, porque permiten mejorar la eficiencia del algoritmo al reducir el número de verificaciones de intersección de triángulos y de iteraciones sobre el arreglo B .

Al ejecutar el Algoritmo 2 sobre el tipo de orden correspondiente a un conjunto de 8 puntos en posición convexa, el número total de nodos explorados es 128,711. En comparación con el Algoritmo 1 el número de nodos es aproximadamente 600 veces menor, lo que representa una reducción significativa. De hecho, la reducción en el tiempo de ejecución es suficiente para permitir encontrar la altura del árbol para el conjunto de 9 puntos en posición convexa. Ésta altura es 10, y la ejecución del algoritmo tardó tan solo 388.49 milisegundos recorriendo 5,135,253 nodos en total. Suponiendo que los demás tipos de orden tardan aproximadamente lo mismo, esto da una estimación de aproximadamente 17 horas para encontrar el valor de T_9 .

El Algoritmo 2 no se ve afectado por el orden de los triángulos en el

Algoritmo 2: Recorre completamente el árbol de búsqueda.

Input: Conjunto S de n puntos.
Output: Altura h del árbol de búsqueda, es decir T_n .

```

1 Function Encuentra-Thrackles( $S$ )
2    $A \leftarrow$  arreglo con los  $\binom{n}{3}$  triángulos en  $S$ .
3    $B \leftarrow$  arreglo con 1 o 0 de tamaño  $\binom{n}{3}$ .
4    $C_1, \dots, C_l \leftarrow$  Cada  $C_i$  es un conjunto de enteros vacío.
5    $T[0] \leftarrow A[0]$ 
6    $t \leftarrow 0$           /* Índice del último triángulo elegido. */
7    $h \leftarrow 0$           /* Altura final del árbol. */
8    $l \leftarrow 1$           /* Nivel actual en el árbol. */
9   while  $l > 0$  do
10    for cada  $i$  tal que  $B[i] = 1$  y  $t < i$  do
11      if  $A[i]$  tiene aristas en común con  $A[t]$  then
12         $B[i] \leftarrow 0$  y agrega  $i$  a  $C_l$ .
13        continue
14      if  $A[i]$  no intersecta a  $A[t]$  then
15         $B[i] \leftarrow 0$  y agrega  $i$  a  $C_l$ .
16    if Existe  $j$  tal que  $B[j] = 1$  y  $t < j$  then
17       $t \leftarrow$  Menor  $j$  tal que  $B[j] = 1$  y  $t < j$ .
18    else
19       $t \leftarrow$  NULL.
20    Loop
21      if  $t \neq$  NULL then
22         $T[l] \leftarrow A[t]$ 
23         $l \leftarrow l + 1$ 
24         $h \leftarrow \text{máx}(h, l)$ 
25        break
26      else
27         $l \leftarrow l - 1$ 
28        if  $l \geq 0$  then
29           $t \leftarrow$  Índice del triángulo  $T[l]$  en  $A$ .
30           $B[j] \leftarrow 1$  Para cada  $j$  en  $C_l$  con  $t < j$ .
31           $C_l \leftarrow \emptyset$ 
32        else
33          break
34    return  $h$ 

```

arreglo A a diferencia del Algoritmo 1 en el que utilizar un orden aleatorio reducía el tiempo de ejecución. Al recorrer el árbol por completo sin importar el orden de los triángulos, no solo se recorre el mismo número total de nodos sino que además el número de nodos por nivel es el mismo.

Al ejecutar el Algoritmo 2 para cada conjunto de 9 puntos, se obtuvo que cada tipo de orden tiene un thrackle de 10 triángulos, y que este es el máximo tamaño posible. La ejecución tardó 9 horas y 57 minutos, y recorrió 2,526,172.5 nodos en promedio. El tiempo final representa un poco más del 50% del tiempo que habíamos estimado, lo que hace pensar que para tipos de orden que no están en posición convexa el algoritmo es más rápido en promedio (el árbol tiene menos nodos). Todo esto prueba computacionalmente el siguiente teorema.

Teorema 11. *Para $n = 9$, $T_n = 10$.*

Para el conjunto de 10 puntos en posición convexa la ejecución del Algoritmo 2 tarda aproximadamente 14 segundos en explorar todo el árbol de búsqueda, que tiene 305,532,771 nodos en total (8 GiB si lo intentáramos almacenar en RAM), y muestra que el thrackle de tamaño máximo tiene 12 triángulos. Suponiendo que los demás tipos de orden tienen un comportamiento similar, hallar el valor exacto de T_{10} con el mismo método que se usó para determinar T_9 , tardaría aproximadamente 8 años. Aunque no disponemos del tiempo necesario para realizar esta prueba, podemos dar una cota inferior para el tamaño de un thrackle maximal sobre un conjunto de 10 puntos.

En la implementación de los Algoritmos 1 y 2 los puntos de S son un arreglo que mantiene el orden de los puntos en la base de datos. Sea $S_1 = [p_1, \dots, p_n]$ el arreglo correspondiente a un conjunto de puntos, cada triángulo $t_1 = \{p_a, p_b, p_c\}$ sobre S_1 puede asociarse a un triángulo en otro conjunto de puntos $S_2 = [q_1, \dots, q_n]$ por medio de los índices, de modo que a t_1 le corresponde el triángulo $t_2 = \{q_a, q_b, q_c\}$. Sea T_1 un thrackle de triángulos en S_1 , y T_2 el conjunto de triángulos que resulta de asociar cada triángulo en T_1 con triángulos en S_2 . En la práctica observamos que frecuentemente T_2 es también un thrackle. Como verificar que un conjunto de triángulos es un thrackle toma menos tiempo que correr el Algoritmo 2, para hacer la búsqueda un poco más rápida, una vez encontrado algún thrackle sobre un conjunto S_i , se verifica si es también un thrackle en S_{i+1} , en caso contrario se corre el Algoritmo 2 sobre S_{i+1} . Este procedimiento da sólo una cota inferior pues encontrar un thrackle de cierto tamaño para todo tipo de orden con $n = 10$, no prueba que no exista algún tipo de orden con un thrackle de mayor tamaño.

Como verificamos que el conjunto de 10 puntos en posición convexa tiene un thrackle de tamaño 12, supusimos que igual a los casos anteriores, el tamaño máximo de un thrackle sería el mismo para conjuntos de puntos en posición no convexa. Ejecutamos el Algoritmo 1 con $n = 10$ y $k = 12$, utilizando un orden aleatorio de triángulos. La prueba tardó aproximadamente 30 horas, recorriendo en promedio 1,768,196 nodos en cada ejecución. De los 14,309,547 tipos de orden el Algoritmo 1 sólo se ejecutó 1,272,248 (8.9%) veces gracias a la estrategia descrita en el párrafo anterior. La razón por la que tomó tanto tiempo es que aproximadamente 79,712 tipos de orden (el 0.56% del total) no tienen thrackles de tamaño 12 como habíamos supuesto en un principio, es decir que en estos casos el algoritmo tuvo el peor tiempo de ejecución y recorrió el árbol de búsqueda por completo. Afortunadamente estos tipos de orden tienen árboles de búsqueda relativamente pequeños (aproximadamente 50 millones de nodos) y sólo toma 1-2 segundos explorarlos. A diferencia de los 305 millones de nodos y 12-14 segundos que tarda el Algoritmo 1 para el conjunto de 10 puntos en posición convexa.

Como los resultados de la ejecución anterior sólo prueban que el tamaño de un thrackle maximal sobre un conjunto de 10 puntos no es 12 para todos los tipos de orden, esto no nos da la cota inferior que buscamos (alguno de los tipos de orden sin thrackle de tamaño 12, podría tener como tamaño máximo de thrackle sea menos de 11). Para obtenerla ejecutamos nuevamente el Algoritmo 1 con $n = 10$ y $k = 11$, bajo la suposición de que el 0.56% de tipos de orden sin thrackles de tamaño 12, si tienen de tamaño 11. Tras 35 minutos y 22 segundos la ejecución terminó exitosamente, recorriendo 98,371.48 nodos en promedio en las 535,383 ejecuciones hechas del Algoritmo 1. Lo que prueba computacionalmente el siguiente teorema.

Teorema 12. *Todo conjunto de 10 puntos en posición general tiene al menos un thrackle de triángulos de tamaño 11.*

Para probar que $T_n = 12$ para $n = 10$, haría falta comprobar que ningún tipo de orden tiene un thrackle de más de 12 triángulos. Esto continuaría dando evidencias de que la Conjetura 5 es cierta. Pese a que en este momento no disponemos del tiempo ni los recursos computacionales, creemos que con con mas recursos computacionales, y paralelizando la ejecución sobre la base de datos, es posible obtener este valor.

4.2. Número de thrackles distintos

Además de estudiar el valor de T_n , utilizando el Algoritmo 2 analizamos el número mínimo de iteraciones que debe hacer un algoritmo que itera

sobre todos los thrackles de triángulos. Para esto contamos cuántos thrackles tiene cada tipo de orden e identificamos cuáles son los que tienen la mayor cantidad. Para conjuntos de n puntos con $n \leq 8$ todos los tipos de orden tienen el mismo número de thrackles de triángulos. Los valores se muestran en la Tabla 4.1.

n	3	4	5	6	7
Número de thrackles	1	4	15	30	30

Tabla 4.1: Número de thrackles de triángulos para conjuntos n puntos con $n \leq 8$.

Para n más grande, los tipos de orden ya no tienen el mismo número de thrackles, analizamos sólo los que tienen el mayor número de thrackles. Las Figuras 4.1 y 4.2 muestran los conjuntos de 8 y 9 puntos con mayor número de thrackles maximales. A partir de estos resultados planteamos la siguiente conjetura:

Conjetura 6. *Los conjuntos de puntos con mayor número de thrackles distintos son conjuntos 2-convexos.*

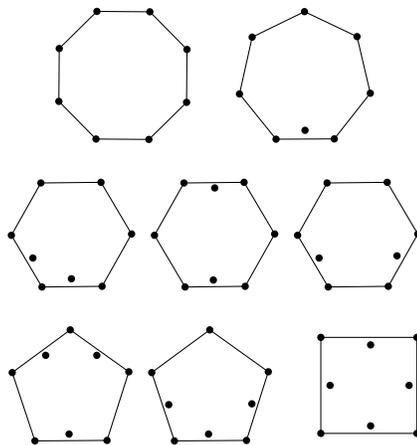


Figura 4.1: Conjuntos de 8 puntos con mayor número de thrackles de triángulos. Cada uno tiene 120 thrackles.

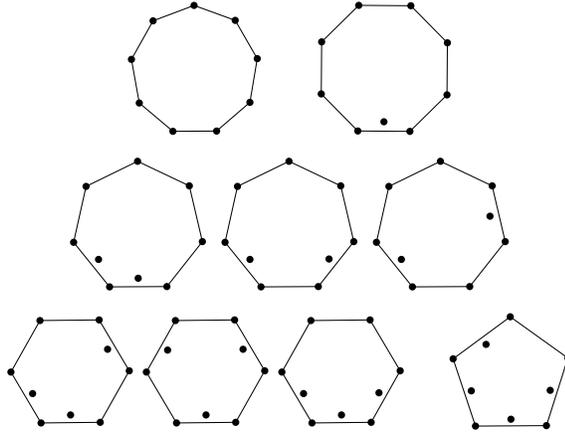


Figura 4.2: Conjuntos de 9 puntos con mayor número de thrackles de triángulos. Cada uno tiene 3156 thrackles.

Capítulo 5

Puntos en posición convexa

En este capítulo presentamos los resultados obtenidos de la búsqueda de thrackles de triángulos sobre conjuntos de n puntos en posición convexa. Al ser un caso particular con más restricciones que el trabajo del capítulo anterior, es posible obtener resultados más concretos.

El capítulo se divide en tres secciones, la primera describe una construcción de un thrackle para todo n que muestra los detalles cuando $3 \nmid n$. La segunda presenta una cota inferior del número de thrackles distintos que genera esta construcción. Finalmente introducimos formas de ordenar los triángulos, para mejorar el tiempo de ejecución del Algoritmo 1, presentado en el capítulo anterior.

5.1. Cota inferior del tamaño de un thrackle

Presentamos una construcción de un thrackle de triángulos sobre un conjunto de puntos en posición convexa, para cada n que busca generar un thrackle de tamaño máximo. Pese a que parece difícil mejorar esta cota, no se ha probado que sea también una cota superior. A continuación presentamos algunos resultados que serán usados en la sección.

Lema 1. *Sea $S = \{p_1, \dots, p_n\}$ un conjunto de n puntos en posición convexa etiquetados en sentido horario. Sean $p_a p_b$ y $p_c p_d$ dos segmentos de recta con $a, b, c, d \in \{1, \dots, n\}$, $a \neq b$ y $c \neq d$. Si $a < c < b < d$ entonces los segmentos $p_a p_b$ y $p_c p_d$ se cruzan. Si $a \leq c \leq b < d$ entonces los segmentos $p_a p_b$ y $p_c p_d$ se intersectan.*

Demostración. Como $a < c < b$ y el conjunto de puntos está en posición convexa, el ángulo $p_b p_a p_c$ es a lo sumo π , entonces el punto p_c está en el

semiplano izquierdo de la recta dirigida definida por $p_a p_b$. Igualmente, el ángulo $p_a p_b p_d$ es a lo sumo π por lo que el punto p_d está en el semiplano derecho de la recta dirigida definida por $p_a p_b$. Esto quiere decir que los puntos p_c y p_d están en semiplanos diferentes de la recta definida por $p_a p_b$. Por el mismo argumento los puntos p_a y p_b están en semiplanos diferentes de la recta definida por $p_c p_d$. Como el segmento $p_a p_b$ divide a los puntos p_c y p_d y el segmento $p_c p_d$ divide a los puntos p_a y p_b entonces ambos segmentos se cruzan.

Si permitimos que $c = a$ o $c = b$ entonces sólo podemos garantizar que los segmentos se intersectan. \square

El Teorema 13 da las condiciones necesarias para la existencia de una 1-factorización en una gráfica bipartita, fue tomado de [7].

Teorema 13 ([7]). *Toda gráfica bipartita r -regular con $r \geq 1$ tiene una 1-factorización.*

La existencia de emparejamientos en gráficas completas se deriva del Teorema 14, igualmente tomado de [7]. De este teorema se sigue el Corolario 1 de forma directa.

Teorema 14 ([7]). *Para cada entero positivo n , la gráfica completa K_{2n} es 1-factorizable.*

Corolario 1. *Toda gráfica completa con un número par de vértices tiene al menos un emparejamiento perfecto.*

El Teorema 15 es el principal resultado de esta sección. La demostración se basa en la construcción de un thrackle de tamaño $\frac{n^2}{9} - O(n)$ presentada en [6], agregando los detalles de los casos de conjuntos de n puntos en los que $3 \nmid n$.

Teorema 15. *Para todo conjunto de n puntos en el plano en posición convexa con $n \geq 6$, existe un thrackle de triángulos, cuyo tamaño k como función de n es:*

$$k(n) = \begin{cases} \frac{n^2}{9} + 1 & \text{si } 3|n \\ \frac{n^2}{9} - \frac{n}{18} + \frac{35}{18} & \text{si } 6|n - 1 \\ \frac{n^2}{9} - \frac{n}{18} + \frac{13}{9} & \text{si } 6|n - 4 \\ \frac{n^2}{9} - \frac{n}{9} + \frac{7}{9} & \text{si } 6|n - 5 \\ \frac{n^2}{9} - \frac{n}{9} + \frac{16}{9} & \text{si } 6|n - 2 \end{cases}$$

Demostración. Existen tres casos principales, cuando $3|n$, $3|n-1$ y $3|n-2$, los últimos dos a su vez se dividen en otros dos casos dependiendo de si $\frac{n}{3}$ es par o impar. Estos 5 casos cubren todos los residuos módulo 6 de los enteros. Para cada uno de ellos construiremos un thrackle de tamaño $k(n)$.

Sea S un conjunto de n puntos en posición convexa. Etiquetamos los elementos de S en sentido horario $S = \{p_1, \dots, p_n\}$. Para el caso $3|n$ definimos los conjuntos

$$\begin{aligned} A &= \{p_1, \dots, p_{n/3}\}, \\ B &= \{p_{n/3+1}, \dots, p_{2n/3}\} \text{ y} \\ C &= \{p_{2n/3+1}, \dots, p_n\} \end{aligned}$$

todos ellos con $\frac{n}{3}$ puntos.

El conjunto $B \cup C$ de vértices y las aristas que tienen un vértice en B y un vértice en C , forman una gráfica bipartita completa que denotamos como $K_{\frac{n}{3}, \frac{n}{3}}(B \cup C)$ o simplemente $K_{\frac{n}{3}, \frac{n}{3}}$. Observemos que $K_{\frac{n}{3}, \frac{n}{3}}$ es $\frac{n}{3}$ -regular. Como $\frac{n}{3} \geq 1$ y $K_{\frac{n}{3}, \frac{n}{3}}$ es bipartita, por el Teorema 13, $K_{\frac{n}{3}, \frac{n}{3}}$ tiene una 1-factorización \mathcal{F} . Sea $m \in \mathcal{F}$ un 1-factor, como los conjuntos A , B y C son disjuntos, con cada arista de m y el vértice $p_1 \in A$ formamos $\frac{n}{3}$ triángulos. Como hay $\frac{n}{3}$ 1-factores en \mathcal{F} se construye un total de $(\frac{n}{3})^2$ triángulos, que a continuación probaremos que forman un thrackle.

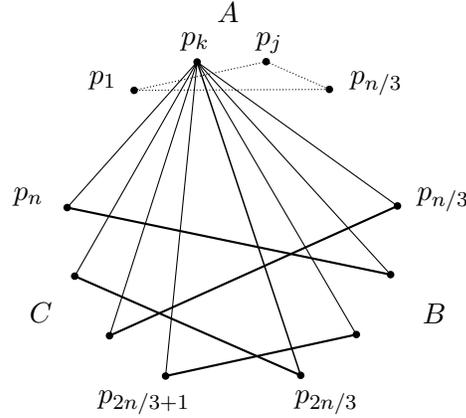
Sea $p_i \in A$, en p_i inciden $\frac{n}{3}$ triángulos, cada uno con exactamente un vértice en B y uno en C . Los triángulos que inciden en p_i se intersectan pues tienen a p_i en común. Sean $p_i, p_j \in A$ con $i < j$, $p_b \in B$ y $p_c \in C$, las aristas $p_i p_b$ y $p_j p_c$ pertenecen a algún triángulo incidente en p_i y p_j respectivamente. Como $i < j < p_b < p_c$ por el Lema 1, las aristas $p_i p_b$ y $p_j p_c$ se cruzan.

Todavía es posible agregar un triángulo más al thrackle. Como cada triángulo tiene una arista $p_a p_c$ con $p_a \in A$ y $p_c \in C$, ésta intersecta a la arista $p_1 p_{n/3}$ por el Lema 1, ya que $1 \leq a \leq n/3 < c$. Es decir que podemos formar el triángulo $p_1 p_a p_{n/3}$ con $p_a \in A$. Todo esto prueba el primer caso, es decir que si $3|n$ entonces existe un thrackle con $\frac{n^2}{9} + 1$ triángulos. La figura 5.1 muestra la construcción.

Para el segundo caso $3|n-1$, definimos los conjuntos

$$\begin{aligned} A &= \{p_1, \dots, p_{(n-1)/3}\}, \\ B &= \{p_{(n-1)/3+1}, \dots, p_{2(n-1)/3}\} \text{ y} \\ C &= \{p_{2(n-1)/3+2}, \dots, p_n\} \end{aligned}$$

al punto restante lo denotamos como $p_x = p_{2(n-1)/3+1}$. Al igual que en el caso anterior, utilizamos estos tres conjuntos para construir $(\frac{n-1}{3})^2$ triángulos que

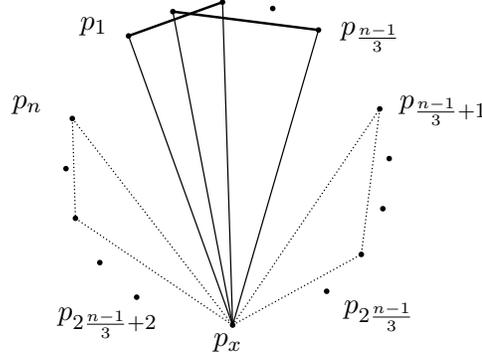
Figura 5.1: Caso $3|n$.

se intersectan dos a dos. Para esto tomamos emparejamientos perfectos de la gráfica bipartita completa con bipartición $B \cup C$, y los asociamos a vértices en A .

Sea $K_{\frac{n-1}{3}}$ la gráfica completa formada por los vértices del conjunto A . Si $\frac{n-1}{3}$ es par, por el Lema 1 existe un emparejamiento perfecto en $K_{\frac{n-1}{3}}$ con $\frac{n-1}{6}$ aristas. Si $\frac{n-1}{3}$ es impar eliminamos un vértice de $K_{\frac{n-1}{3}}$ y formamos un emparejamiento con $\frac{(n-1)/3-1}{2} = \frac{n-4}{6}$ aristas. En ambos casos con cada arista del emparejamiento formamos un triángulo utilizando el vértice p_x . Cada triángulo nuevo tiene dos aristas $p_a p_x$ con $p_a \in A$, mientras que los triángulos anteriores todos tienen una arista $p_b p_c$ con $p_c \in C$ y $p_b \in B$. Como $a < b < x < c$, por el Lema 1 $p_a p_x$ y $p_b p_c$ se cruzan. Entonces, todos estos triángulos forman un thrackle.

A continuación agregamos dos triángulos más que tienen al vértice p_x . Como ambos triángulos tienen a p_x como vértice, entonces se intersectan a todos los generados a partir del emparejamiento. Como los primeros $(\frac{n-1}{3})^2$ triángulos tienen una arista $p_c p_b$ con $p_b \in B$ y $p_c \in C$, por el Lema 1 la arista $p_x p_n$ los intersecta a todos ya que $b < x < c \leq n$. Con la arista $p_x p_n$ y un punto en C formamos un nuevo triángulo que intersecta a todos los demás. Igualmente con la arista $p_x p_{(n-1)/3+1}$ y un punto en B formamos otro triángulo. La construcción final se muestra en la figura 5.2. Al final, el tamaño del thrackle es el siguiente:

$$\begin{aligned} \left(\frac{n-1}{3}\right)^2 + \frac{n-1}{6} + 2 &= \frac{n^2}{9} - \frac{n}{18} + \frac{35}{18} && \text{si } 6|n-1 \\ \left(\frac{n-1}{3}\right)^2 + \frac{n-4}{6} + 2 &= \frac{n^2}{9} - \frac{n}{18} + \frac{13}{9} && \text{si } 6|n-4 \end{aligned}$$

Figura 5.2: Caso $3|n - 1$.

Para el último caso, se divide el conjunto de puntos de la siguiente forma

$$\begin{aligned} A &= \{p_1, \dots, p_{(n-2)/3}\}, \\ B &= \{p_{(n-2)/3+1}, \dots, p_{2(n-2)/3}\} \text{ y} \\ C &= \{p_{2(n-1)/3+3}, \dots, p_n\} \end{aligned}$$

denotando los dos puntos restantes como $p_x = p_{2(n-2)/3+1}$ y $p_y = p_{2(n-2)/3+2}$. Al igual que en los casos anteriores, creamos $(\frac{n-2}{3})^2$ triángulos entre los conjuntos A , B , y C , a partir de emparejamientos entre B y C .

Suponiendo que $|A|$ es impar, se puede formar un thrackle de $|A|$ segmentos de tamaño máximo que forma un ciclo, y en el que cada vértice tiene grado dos. Recorriendo el ciclo formado por este thrackle, se pueden colorear las aristas alternando 2 colores, excepto por la última. En esta coloración no hay dos aristas del mismo color que compartan un vértice, por lo que podemos formar triángulos disjuntos en aristas con los segmentos de un color y p_x , y con los del otro color y p_y . Estos triángulos se intersectan entre sí porque los segmentos del thrackle se intersectan, e intersectan a todos los demás triángulos porque todas las aristas $p_a p_x$ y $p_a p_y$ con $p_a \in A$ intersectan a la arista $p_b p_c$ de los triángulos formados hasta ahora. En total, esto agrega $\frac{n-2}{3} - 1 = \frac{n-5}{3}$ triángulos nuevos, cuando $6|n - 5$. La figura 5.3 muestra estos $\frac{n-5}{3}$ triángulos.

Si $|A|$ es par, se utiliza un thrackle de segmentos en A que no incluye al punto p_1 . Al igual que antes se usan dos colores para colorear las aristas. Se inicia la coloración en $p_{(n-2)/3}$ asociando el primer color a p_x de tal forma que la arista del thrackle que no se utiliza sea $p_i p_{(n-2)/3}$ con $1 < i < (n-2)/3$. Si consideramos los triángulos $p_1 p_x p_i$ y $p_1 p_y p_{(n-2)/3}$, estos son disjuntos en aristas con los demás formados con p_x y p_y porque las aristas $p_1 p_i$ y $p_1 p_{(n-2)/3}$ no son parte del thrackle. Además el triángulo formado con el

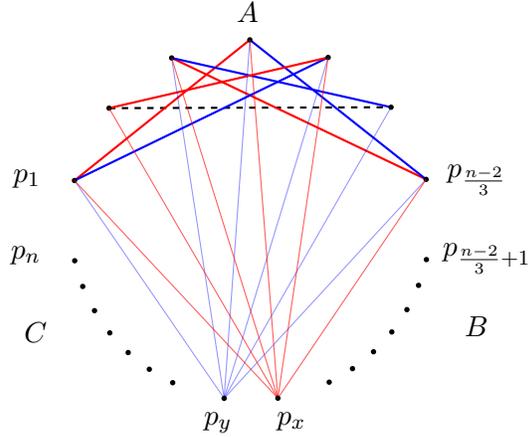


Figura 5.3: $\frac{n-5}{3}$ triángulos adicionales para $3|n - 2$ con $|A|$ impar (no se incluyen los triángulos entre los conjuntos B y C).

thrackle que contiene a p_i está asociado a p_y mientras que el que usa a $p_{(n-2)/3}$ se conecta a p_x , es decir que las aristas $p_i p_x$ y $p_{(n-2)/3} p_y$ no habían sido utilizadas antes. Estos dos triángulos se intersectan entre ellos porque comparten a p_1 como vértice. El triángulo $p_1 p_y p_{(n-2)/3}$ se intersecta con todos los demás porque la arista $p_y p_{(n-2)/3}$ separa a A de B y todo triángulo tiene una arista que $p_a p_b$, excepto en el caso de los triángulos que contienen a $p_{(n-2)/3}$ pero estos comparten un vértice por lo que también lo intersectan. La arista $p_x p_1$ se comporta de forma análoga en el triángulo $p_1 p_x p_i$. Al final en este caso cuando $6|n - 2$, se obtienen $\frac{n-2}{3}$ triángulos adicionales.

Independientemente de la paridad de $|A|$ es posible agregar otros dos triángulos a la construcción. El primero utiliza la arista $p_x p_n$ y cualquier otro vértice en C , puesto que todo triángulo formado hasta el momento contiene a p_x , a p_y o un punto p_c , y como $p_x < p_y < p_c$ todos los triángulos que no se forman con p_n ni con p_x la intersectan, los demás comparten un vértice con la arista. Análogamente la arista $p_y p_{(n-2)/3+1}$ con algún vértice de B distinto de $p_{(n-2)/3+1}$ forma otro triángulo que intersecta a todos los demás. Estos dos triángulos se intersectan mutuamente porque las aristas que los forman igualmente se intersectan. La figura 5.4 muestra el thrackle final cuando $|A|$ es par. El thrackle final para cuando $|A|$ es impar es igual a la figura 5.3 agregando dos triángulos descritos en este párrafo.

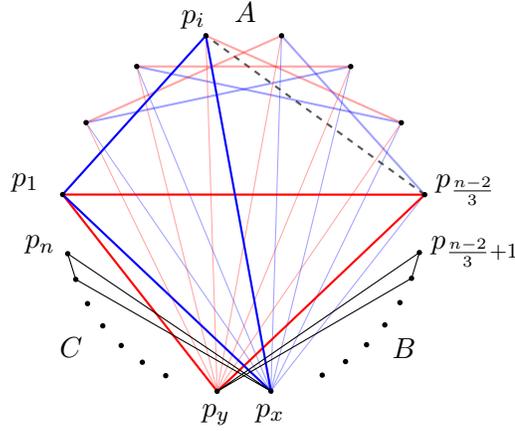


Figura 5.4: Caso $3|n - 2$ con $|A|$ par (no se incluyen los triángulos entre los conjuntos B y C).

En resumen, para este caso el número total de triángulos en el thrackle es:

$$\left(\frac{n-2}{3}\right)^2 + \frac{n-5}{3} + 2 = \frac{n^2}{9} - \frac{n}{9} + \frac{7}{9} \text{ si } 6|n-5$$

$$\left(\frac{n-2}{3}\right)^2 + \frac{n-2}{3} + 2 = \frac{n^2}{9} - \frac{n}{9} + \frac{16}{9} \text{ si } 6|n-2$$

□

5.2. Número de thrackles distintos

A partir de esta construcción de un thrackle con muchos triángulos estudiamos el número de thrackles distintos que hay sobre un conjunto de puntos en posición convexa. La construcción del thrackle propuesta en el Teorema 15 siempre toma una 1-factorización de una gráfica bipartita completa con partes de tamaño $\lfloor \frac{n}{3} \rfloor$, esto quiere decir que si contamos cuántas 1-factorizaciones tiene en general una gráfica bipartita completa $K_{n,n}$, este valor estará directamente relacionado con el número de thrackles distintos para un conjunto de puntos en posición convexa. A continuación introducimos los conceptos que utilizaremos para acotar el número de 1-factorizaciones de $K_{n,n}$.

Un *multiconjunto* es un conjunto que permite repeticiones de sus elementos. Sea A un multiconjunto, denotamos como m_A a su *función de multiplicidad*, de forma que para cada $a \in A$, $m_A(a)$ representa cuántas veces aparece

a en A . Le llamamos *conjunto base* de A al conjunto de sus elementos sin repeticiones, y lo denotamos como $B(A)$. Si denotamos la cardinalidad de A como $|A|$, observamos que $|A| = \sum_{x \in B(A)} m_A(x)$.

Sea n un entero positivo, una *partición entera de n* es una representación de n como sumas de enteros. Formalmente definimos una *partición entera de n* como un multiconjunto λ . Los elementos de λ suman n y son llamados *partes* de λ . Denotamos una partición λ de n como $\lambda \vdash n$. Denotamos además al conjunto de todas las particiones enteras de n que no contienen al 1 como

$$P'(n) = \{A : \forall a \in A, \quad 2 \leq a \leq n \quad \text{y} \quad \sum_{x \in B(A)} m_A(x) \cdot x = n\}$$

Para un entero n , definimos a $p(n)$ como el número de particiones enteras de n . Observemos que $p(n) \geq |P'(n)|$

Lema 2. *Sea n un entero positivo, entonces*

$$|P'(n)| = p(n) - p(n-1).$$

Demostración. Sea λ una partición de n si una de las partes de λ es igual a 1, entonces λ es de la forma: $\{1\} \cup \lambda' \vdash (n-1)$ para alguna partición λ' . El número total de particiones de esta forma es igual al número total de particiones λ' . Esto significa que existe una biyección entre las particiones de $n-1$ y las particiones que contienen al 1 de n . Como queremos las que no contienen al 1, restamos este valor del total de particiones obteniendo $p(n) - p(n-1)$. \square

Sea F un 2-factor de la gráfica bipartita completa $K_{n,n}$, denotamos al conjunto de subgráficas de F que son sus componentes conexas como $\mathcal{F} = \{C^1, \dots, C^k\}$. A continuación utilizaremos la abreviación $v(C^i)$ para referirnos a $|V(C^i)|$ (número de vértices de la subgráfica C^i). Por el Teorema 2 cada C^i es un ciclo. Por el Teorema 1, $v(C^i)$ debe ser par. A cada 2-factor F le asociamos un multiconjunto A que representa los tamaños de los ciclos en F . A este multiconjunto A le llamamos *multiconjunto característico* de F y lo definimos formalmente como

$$A = \{a : a = v(C^i)/2 \quad \text{para} \quad C^i \in \mathcal{F}\}.$$

Observamos a continuación que A es una partición entera de n con partes diferentes a 1, es decir que $A \in P'(n)$. Para cada C_i , $v(C^i) \geq 4$, y tiene $v(C^i)/2$ vértices en cada conjunto de la bipartición de $K_{n,n}$, por lo que

$$\sum_{C^i \in \mathcal{F}} \frac{v(C^i)}{2} = n.$$

Definimos la función $L: P'(n) \rightarrow \mathbb{N}$ tal que $L(A)$ es el número de 2-factores que tienen a A como multiconjunto característico. Es decir que $L(A)$ representa el número de 2-factores de $K_{n,n}$ con componentes conexas del mismo tamaño. Cuando $A = \{n\}$, $L(A)$ representa de cuántas formas el ciclo C^i con $v(C^i) = 2n$ es subgráfica de $K_{n,n}$, a este número lo denotamos como e_n . La Figura 5.5 muestra los 2-factores que cuenta el valor de e_n para $n = 3$.

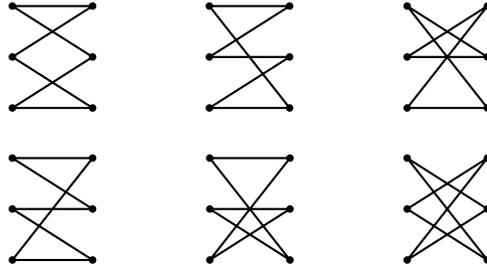


Figura 5.5: 2-factores de la gráfica bipartita completa $K_{3,3}$, con multiconjunto característico $\{3\}$. Esto ilustra que $e_3 = 6$

Lema 3. Para todo entero positivo n

$$e_n = \frac{n!^2}{2n}$$

Demostración. Este problema es equivalente a tomar la gráfica C_{2n} y etiquetar sus vértices, primero de forma alternada con dos colores A y B, para marcar la bipartición y después cada color con las posibles permutaciones de $\{1, \dots, n\}$, este procedimiento da un total de $n!^2$ subgráficas. Para cada etiquetación de C_{2n} existen n equivalentes que son sólo traslaciones de la misma etiquetación moviendo cada etiqueta 2 posiciones en el C_{2n} . Además a cada cada subgráfica le pertenecen 2 etiquetaciones dependiendo de la dirección con que se recorra el C_{2n} . Todo esto genera un sobreconteo de $2n$ de donde se deriva la expresión de e_n . \square

Conociendo que $e_2 = 1$, esta expresión también se puede derivar de la relación de recurrencia

$$e_n = 2e_{n-1} \binom{n}{2} = 2^{n-2} \prod_{i=2}^n \binom{i}{2} = \frac{n!^2}{2n}$$

donde se puede ver de forma más clara que $\binom{n}{2} | e_n$.

Lema 4. Para $A \in P'(n)$, con $A = \{a_1, \dots, a_k\}$ tal que si $i \neq j$ entonces $a_i \neq a_j$. Denotamos como S_x para $1 \leq x \leq k$ a la suma de los elementos de A con subíndice menor a x , es decir $S_x = \sum_{z=1}^{x-1} a_z$. Entonces

$$L(A) = \prod_{x=1}^k e_{a_x} \binom{n - S_x}{a_x}^2$$

Demostración. Recordemos que $L(A)$ representa cuántos 2-factores de $K_{n,n}$ tienen como componentes conexas ciclos con los mismos tamaños (representados por A). A continuación contamos de cuantas formas es posible elegir cada componente conexa como subgráfica de $K_{n,n}$. Sea F un 2-factor de la gráfica bipartita completa $K_{n,n}$ con multiconjunto característico A , y sea C_i una componente conexa de F con $2a_i$ vértices donde $a_i \in A$. El valor de e_{a_i} representa el número de formas distintas de elegir C_i como subgráfica de la gráfica bipartita completa K_{a_i, a_i} . Además, el número de formas de elegir K_{a_i, a_i} como subgráfica de $K_{n,n}$ está dada por $\binom{n}{a_i}^2$. Entonces, el total de formas de elegir C_i como subgráfica de $K_{n,n}$ es $e_{a_i} \binom{n}{a_i}^2$.

Sea C_j una componente conexa de F distinta a C_i , con $2a_j$ vértices y $a_j \in A$. Por cada forma de elegir C_i como subgráfica de $K_{n,n}$ hay $e_{a_j} \binom{n-a_i}{a_j}^2$ formas de elegir a C_j porque ya no contamos cuántas formas hay de elegir K_{a_i, a_i} como subgráfica de $K_{n,n}$, sino cuántas formas hay de elegir K_{a_j, a_j} como subgráfica de la gráfica bipartita completa $K_{n-a_i, n-a_i}$. En general S_x representa el número de vértices que ya han sido utilizados de elegir las componentes conexas anteriores. \square

Lema 5. Para $A \in P'(n)$, con $A = \{a_1, \dots, a_k\}$ donde todo $a_i \in A$ tiene el mismo valor a . Entonces

$$L(A) = \frac{e_a^k \prod_{x=1}^k \binom{n-a \cdot x}{a}^2}{k!}$$

Demostración. Este teorema tiene una demostración similar al Teorema 4. Deseamos contar cuántos 2-factores tienen k ciclos C_i como componentes conexas, cada ciclo con $2a$ vértices. El número de formas de elegir un ciclo C_i como subgráfica de $K_{a,a}$ es e_a . El número de formas de elegir $K_{a,a}$ como subgráfica de $K_{n,n}$ es $\binom{n}{a}^2$. Si elegimos sucesivamente cada ciclo C_i , la gráfica bipartita de donde es posible elegirlos va reduciendo el número de vértices en $2a$ por cada selección. La cuenta hasta el momento para el i -ésimo ciclo es $e_a \binom{n-a \cdot i}{a}^2$. Al contar todas las posibles formas de tomar a $K_{a,a}$ como subgráfica de $K_{n,n}$ hay un sobreconteo porque como todos los ciclos son del

mismo tamaño, cambiar el orden no genera un 2-factor distinto, como hay $k!$ permutaciones de los ciclos, la cuenta final para el i -ésimo ciclo es:

$$\frac{e_a \binom{n-a \cdot i}{a}^2}{k!},$$

de donde se obtiene la expresión del teorema. \square

Lema 6. Para $A \in P'(n)$, sea $A = \{a_1, \dots, a_k\}$ tal que $a_i \leq a_j$ para $i < j$ denotamos como S_x para $x \in B(A)$ a la cantidad de elementos de A menores a x , es decir $S_x = \sum_{z \in B(A)} y_{z < x} m_A(z)$. Además denotamos $e_x = e_{m_A(x)}$. Entonces

$$L(A) = \prod_{x \in B(A)} \frac{e_x^{m_A(x)} \prod_{i=1}^{m_A(x)} \binom{n - (S_x + m_A(x)i)}{x}^2}{m_A(x)!}$$

Demostración. Esta es una combinación del Lema 4 y el Lema 5. Para cada $A \in P'(n)$ la idea es ir contando el número de formas de elegir $a \in A$ puntos en cada bipartición de la gráfica bipartita completa $K_{n,n}$ y multiplicando por cuántos ciclos distintos se pueden obtener con estos puntos, como se hizo en la demostración del Lema 5. Por cada elección de $2a$ vértices (a vértices en cada bipartición), multiplicamos por e_a que representa el número de ciclos que se pueden obtener como subgráficas de los $2a$ puntos elegidos. Por cada secuencia de k elementos de A repetidos, dividimos entre $k!$ por como hicimos en el Lema 5. El Algoritmo 3 muestra cómo evaluar $L(A)$ mientras que la Figura 5.6 muestra el procedimiento para $A = \{2, 2, 3\}$ y $n = 7$. \square

Lema 7. Sea A el multiconjunto característico de un 2-factor de $K_{n,n}$ con n par, entonces $L(A)$ toma el mínimo valor si $A = \{2, \dots, 2\}$.

Demostración. Observemos que $L(A)$ tiene la forma:

$$e_{x_1} \frac{\binom{n_1}{x_1}^{2h_1}}{h_1!} \times \dots \times e_{x_k} \frac{\binom{n_k}{x_k}^{2h_k}}{h_k!}.$$

Sabemos que $x_i = 2$ minimiza a $e_{x_i} = e_2 = 1$ y a $\binom{n_i}{x_i}^2 = \binom{n_i}{2}^2$. Además h_i se maximiza con el número de repeticiones, y $A = \{2, \dots, 2\}$ tiene el mayor número de repeticiones. \square

Denotamos como \mathcal{D}_n al conjunto de todas las 1-factorizaciones de $K_{n,n}$, recordemos que nuestro objetivo final es determinar $|\mathcal{D}_n|$. Si la bipartición en

Algoritmo 3: Evalúa $L(A)$.

Input: Arreglo A con elementos en orden ascendente, que contiene elementos de algún multiconjunto $A \in P'(n)$. (Denotamos $A.len$ al número de elementos de A).

Output: Valor de $L(A)$.

```

1 Function  $L(A)$ 
2    $S \leftarrow 0$ 
3    $res \leftarrow 1$ 
4    $i \leftarrow 0$ 
5    $n \leftarrow \sum_{a \in A} a$ 
6   while  $i < A.len$  do
7      $k \leftarrow 0$ 
8      $repeated \leftarrow 1$ 
9     do
10       $repeated \leftarrow repeated \times \binom{n - S}{A[i]}$ 
11       $S \leftarrow S + A[i]$ 
12       $k \leftarrow k + 1$ 
13       $i \leftarrow i + 1$ 
14      if  $i = A.len$  then break
15      while  $A[i - 1] = A[i]$ 
16         $repeated \leftarrow repeated \times \frac{e^k A[i-1]}{k!}$ 
17       $res \leftarrow res \times repeated$ 
18  return  $res$ 

```

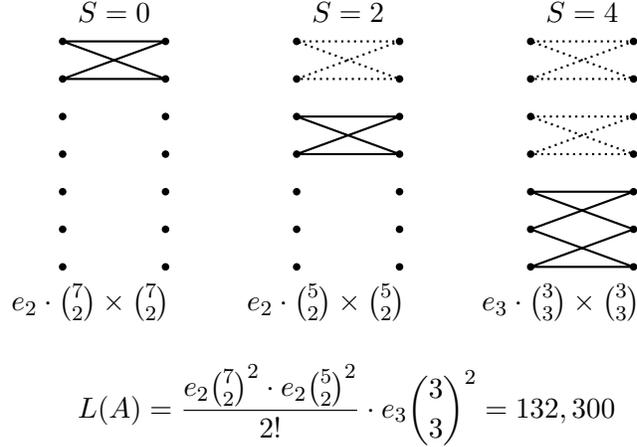


Figura 5.6: Conteo de los 2-factores de una gráfica bipartita completa $K_{n,n}$ para $n = 7$, con multiconjunto característico $A = \{2, 2, 3\}$.

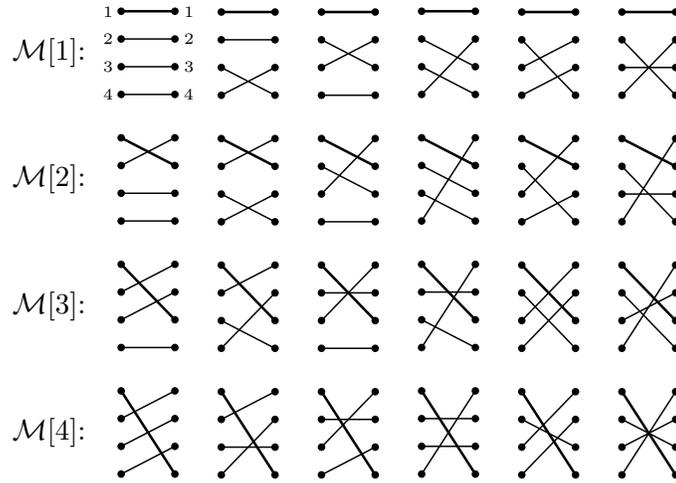
vértices de $K_{n,n}$ es $A \cup B$, etiquetamos estos conjuntos como $A = \{1, \dots, n\}$ y $B = \{1, \dots, n\}$, y nos referimos a las aristas de $K_{n,n}$ como (a, b) con $a \in A$ y $b \in B$. Sea \mathcal{M} el conjunto de todos los 1-factores de $K_{n,n}$, definimos el subconjunto $M[i] \subset \mathcal{M}$ como el conjunto de 1-factores en \mathcal{M} que contienen a la arista $(1, i)$ con $1 \leq i \leq n$. Los conjuntos $M[i]$ son una partición de \mathcal{M} , es decir que son mutuamente disjuntos. La Figura 5.7 muestra el conjunto \mathcal{M} con su partición, para $n = 4$.

Sea \mathcal{T} un árbol con nodos en $\{K_0\} \cup \mathcal{M}$ en el que la raíz es la gráfica vacía K_0 , y los demás nodos son 1-factores de la gráfica bipartita completa $K_{n,n}$ (es decir que los nodos distintos a la raíz pertenecen a \mathcal{M}). Construimos a \mathcal{T} de forma que satisfice las siguientes propiedades:

1. Los nodos de un mismo nivel pertenecen a la misma parte $M[i]$ para algún $1 \leq i \leq n$.
2. Sea $\{K_0, m_1, m_2, \dots, m_i\}$ el camino de la raíz al nodo m_i , el nodo m_{i+1} es hijo de m_i si y sólo si m_{i+1} no comparte ninguna arista con $m_1 \cup m_2 \cup \dots \cup m_i$.

La Figura 5.8 muestra parte del árbol \mathcal{T} para $n = 4$.

Sea $m \in \mathcal{M}$ un nodo de \mathcal{T} distinto a la raíz, observamos que por la propiedad 2, el nodo m no comparte aristas con sus ancestros, es decir que un camino de tamaño n representa una 1-factorización. Por el teorema 13

Figura 5.7: Partición del conjunto \mathcal{M} , para $n = 4$.

esto significa que la altura del árbol \mathcal{T} es exactamente n , y todo camino que va de la raíz a una hoja tiene longitud n .

Lema 8. *Cada 1-factorización de $K_{n,n}$ aparece exactamente una vez como un camino de la raíz a una hoja.*

Demostración. Por contradicción, suponemos que existe una 1-factorización $d_n = \{m_1, \dots, m_n\}$ de $K_{n,n}$ que no aparece como algún camino de \mathcal{T} . Esto significa que para algunos $m_i \in d_n$ y $m_j \in d_n$ con $i \neq j$, m_j no aparece como descendiente de m_i , ni m_i como descendiente de m_j . Supongamos que $m_i \in M[i]$, al ser d_n una descomposición, m_i y m_j no pueden compartir ninguna arista, entonces $m_j \notin M[i]$, supongamos que $m_j \in M[j]$ y que además $M[i]$ representa nodos de un nivel más cercano a la raíz que $M[j]$. Por la propiedad 1 de \mathcal{T} , como m_i y m_j pertenecen a partes diferentes, también pertenecen a niveles diferentes del árbol, ahora bien, por la propiedad 2 como m_j no comparte ninguna arista con $m_k \in d_n$, y en particular no comparte ninguna arista con m_i , entonces m_j debe tener a m_i en algún punto de su camino a la raíz, pero habíamos supuesto que m_i y m_j no pertenecían a un mismo camino. \square

Por el Lema 8 se sigue que el número de hojas de \mathcal{T} es equivalente a $|\mathcal{D}_n|$ (el número de 1-factorizaciones de $K_{n,n}$). Observamos que para todos los caminos $\{K_0, m_1, \dots, m_{n-1}, m_n\}$ que sólo difieren en los últimos nodos m_{n-1} y m_n (los primeros $n - 1$ nodos son iguales), la gráfica $m_{n-1} \cup m_n$ es

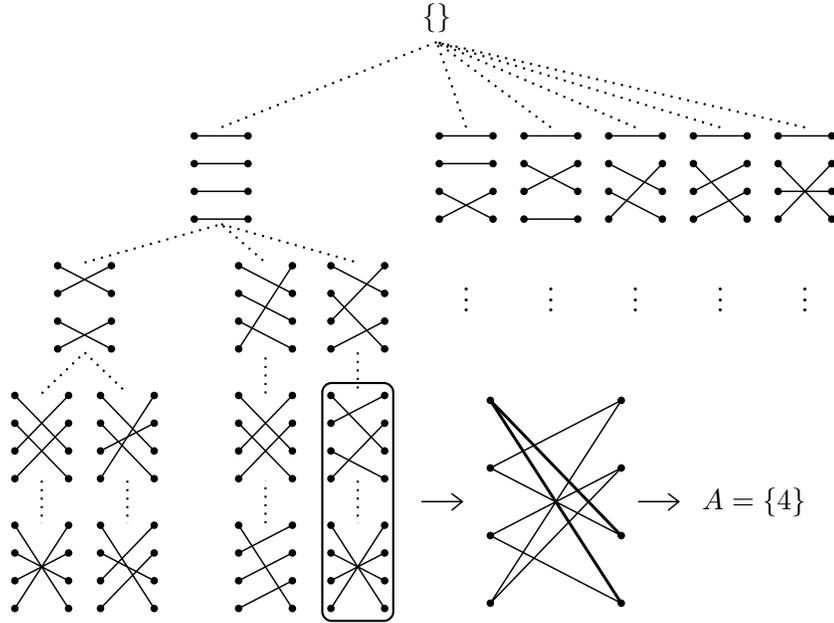


Figura 5.8: Primer subárbol de \mathcal{T} para $n = 4$ y el complemento de uno de los caminos C' con su multiconjunto característico.

siempre la misma subgráfica de $K_{n,n}$ porque debe ser igual a

$$m_{n-1} \cup m_n = K_{n,n} \setminus m_0 \cup \dots \cup m_{n-2}$$

y dijimos que los nodos $m_1 \dots m_{n-2}$ son iguales, a esta gráfica $m_{n-1} \cup m_n$ le llamamos *complemento de C'* donde C' es el camino

$$C' = \{K_0, m_1, \dots, m_{n-2}\}.$$

El complemento de uno de los caminos C' para $n = 4$ se ilustra en la Figura 5.8.

Sea C' un camino de \mathcal{T} de longitud $n - 2$, su complemento es un 2-factor y por lo tanto tiene un multiconjunto característico al que llamamos A . Definimos la función $D_n: P'(n) \rightarrow \mathbb{N}$ tal que $D_n(A)$ representa el número de caminos de longitud $n - 2$ cuyo complemento tiene como conjunto característico a A . Observamos que el número total de nodos en el nivel $n - 2$ (número de caminos C') es igual a $\sum_{A \in P'(n)} D_n(A)$.

Teorema 16. Para todo $A \in P'(n)$ se cumple

$$D_n(A) \geq \frac{L(A)}{\binom{n}{2}}.$$

Demostración. Supongamos que para enteros positivos i y j las partes a las que pertenecen los nodos del penúltimo y el último nivel de \mathcal{T} son $M[i]$ y $M[j]$ respectivamente. El valor de $L(A)$ representa el número de 2-factores que tienen a A como multiconjunto característico, pero todos los complementos correspondientes a cada camino C' de tamaño $n - 2$ contienen las aristas $(1, i)$ y $(1, j)$ por lo que no todo 2-factor es un complemento de C' . Como hay exactamente $\binom{n}{2}$ formas de escoger a i y j de n , el número de 2-factores con estas aristas es $L(A)/\binom{n}{2}$ (recordemos que $\binom{n}{2}|e_n$ por lo tanto $\binom{n}{2}|L(A)$ también).

A continuación probamos que todo 2-factor que contiene a las aristas $(1, i)$ y $(1, j)$ debe aparecer como complemento de al menos un camino parcial de tamaño $n - 2$. Sea F un 2-factor de $K_{n,n}$ que contiene a las aristas $(1, i)$ y $(1, j)$, éste debe ser el complemento de al menos a un camino de longitud $n - 2$ porque al ser una gráfica 2-regular $F = m_i \cup m_j$, y de no existir ese camino, entonces toda 1-factorización que incluye a m_i y m_j no aparecería como camino del \mathcal{T} , pero por el Lema 8 toda 1-factorización es un camino del árbol. \square

Teorema 17. *El número total de 1-factorizaciones de $K_{n,n}$ es*

$$|\mathcal{D}_n| = \sum_{A \in P'(n)} D_n(A) \cdot 2^{|A|-1}.$$

Demostración. Sea $F = m_i \cup m_j$ un 2-factor, cuyo multiconjunto característico es $A = \{a_1, \dots, a_k\}$. Sea C_i el ciclo correspondiente al elemento $a_i \in A$, C_i tiene $2a_i$ vértices. Un ciclo con k vértices tiene siempre dos 1-factores con $k/2$ aristas. Le llamamos m_a y m_b a los 1-factores de C_i . Como F tiene dos 1-factores m_i y m_j , para cada ciclo C_i excepto por el que contiene las aristas $(1, i)$ y $(1, j)$, sus 1-factores tiene dos opciones, ya sea pertenecer a m_j o a m_i . El ciclo que contiene las aristas $(1, i)$ y $(1, j)$ es diferente ya que en este ciclo, 1-factores están determinados por i y j . Al final el total de 1-factorizaciones de F es $2^{|A|-1}$.

Este número representa cuántos hijos tiene el nodo más cercano a las hojas de un camino parcial C' con $n - 2$ nodos, donde C' tiene un complemento con multiconjunto característico A . Es decir que el número total de hojas de \mathcal{T} es exactamente la suma de $D(A) \cdot 2^{|A|-1}$ sobre todos los posibles valores de A . \square

Teorema 18. *Un conjunto de n puntos en posición convexa con $6|n$ tiene al menos*

$$(p(n/3) - p(n/3 - 1)) \frac{\left(\frac{n}{3}\right)! \cdot 6^{\frac{n}{3}-2} \cdot 2^{\frac{n}{6}-1}}{\left(\frac{n}{6}\right)! \binom{n/3}{2}}$$

thrackles de triángulos distintos. Donde $p(n)$ representa el número de particiones enteras de n .

Demostración. Por los teoremas 16 y 17 sabemos que:

$$|\mathcal{D}_{n/3}| = \sum_{A \in P'(n/3)} D_{n/3}(A) \cdot 2^{|A|-1} \geq \sum_{A \in P'(n/3)} \frac{L(A) \cdot 2^{|A|-1}}{\binom{n/3}{2}} \quad (5.1)$$

Como $6|n$, entonces $n/3$ es par, entonces por el Lema 7 $L(A)$ se minimiza cuando $A = \{2, \dots, 2\}$. Utilizando el Lema 5 obtenemos

$$L(A) = \frac{\prod_{j=1}^{n/6-1} \binom{n/3-2j}{2}^2}{n/6!} = \frac{6^{n/3-2}}{n/6!} \quad (5.2)$$

Finalmente de 5.1 y 5.2 obtenemos

$$|\mathcal{D}_{n/3}| \geq \sum_{A \in P'(n/3)} \frac{\frac{6^{n/3-2}}{n/6!} \cdot 2^{n/6-1}}{\binom{n/3}{2}}$$

Por la construcción de thrackles de triángulos de la demostración del Teorema 15, sabemos que habrá $\frac{n}{3}!$ thrackles distintos para cada elección de una 1-factorización de la gráfica bipartita completa $K_{n/3, n/3}$. Por esto, y utilizando el Lema 2 el número de thrackles debe ser al menos:

$$(p(n/3) - p(n/3 - 1)) \frac{\left(\frac{n}{3}\right)! \cdot 6^{\frac{n}{3}-2} \cdot 2^{\frac{n}{6}-1}}{\left(\frac{n}{6}\right)! \binom{n/3}{2}}$$

□

Los resultados de los teoremas anteriores se verificaron experimentalmente utilizando principalmente dos algoritmos. El primero es un algoritmo de backtracking que genera el árbol \mathcal{T} y cuenta la cantidad de hojas (es decir el número de 1-factorizaciones de la gráfica bipartita completa $K_{n,n}$), además del valor de $D_n(A)$ para cada multiconjunto $A \in P'(n)$. El segundo algoritmo genera todos los subconjuntos de $2n$ aristas de $K_{n,n}$ y verifica cuáles de ellos son 2-factores de $K_{n,n}$ y obtiene su multiconjunto característico, de esta forma calculamos el valor de $L(A)$ para cada multiconjunto $A \in P'(n)$.

La Tabla 5.1 muestra la comparación de los valores obtenidos para $n = 6$ y $n = 7$. La primera columna se obtuvo por medio del algoritmo de backtracking mientras que la segunda por la enumeración de todas las subgráficas

2-regulares de $K_{n,n}$ (También se verificó que la fórmula derivada en el Lema 6 genera exactamente los mismos valores en esta columna). En la tabla se puede verificar la relación del Teorema 16 y que además los valores todavía tienen una diferencia significativa, que de hecho es un factor entero en todos los casos. De encontrar una expresión para el valor de esta relación, podríamos encontrar el valor exacto del número de 1-factorizaciones de $K_{n,n}$ en términos de n .

$n = 6$			$n = 7$		
A	$D_n(A)$	$L(A)/\binom{n}{2}$	A	$D_n(A)$	$L(A)/\binom{n}{2}$
{6}	483,840	2,880	{7}	4,727,808,000	86,400
{4, 2}	380,160	1,080	{5, 2}	3,328,819,200	30,240
{3, 3}	184,320	480	{4, 3}	2,748,211,200	25,200
{2, 2, 2}	80,640	90	{3, 2, 2}	1,393,459,200	6,300

Tabla 5.1: Número de caminos de \mathcal{T} de longitud $n - 2$ con complemento A (número de 1-factorizaciones de $K_{n,n}$) y número de 2-factores de $K_{n,n}$ con multiconjunto característico A , para cada $A \in P'(n)$.

5.3. Tamaño de triángulos

En el capítulo anterior presentamos en Algoritmo 1 y mostramos que el orden en el que se encuentran los triángulos al inicio de la ejecución puede afectar el número total de nodos recorridos del árbol de búsqueda. En esta sección presentamos varias formas de ordenar triángulos definidos sobre puntos en posición convexa y mostramos cómo afectan la ejecución del Algoritmo 1.

Sea e una arista de un triángulo definido sobre un conjunto de puntos en posición convexa, e divide al plano en dos. Considérese el número de vértices que quedan en cada semiplano, al menor de ellos le llamamos el *tamaño* de e . Definimos el *tamaño de un triángulo* en un conjunto de puntos en posición convexa como el máximo de los tamaños de sus tres aristas. De la definición podemos observar que el menor tamaño que puede tener un triángulo es 1. y el máximo $\lfloor n/2 \rfloor - 1$. El Algoritmo 4 muestra el código utilizado para calcular el tamaño de un triángulo representado por un arreglo de tres enteros que son etiquetas de los vértices en orden ascendente suponiendo que el conjunto de puntos está etiquetado en sentido horario.

Clasificamos cada thrackle del conjunto de n puntos en posición convexa con $6 \leq n \leq 10$ (para $n < 6$ todos los triángulos tienen el mismo tamaño)

Algoritmo 4: Calcula el tamaño de un triángulo definido sobre un conjunto S de n puntos en posición convexa etiquetados en sentido horario.

Input: Arreglo t con 3 etiquetas en orden ascendente, que forman el triángulo en S

Output: Tamaño del triángulo que representa t .

```

1 Function Tamaño-Triángulo( $n, t$ )
2    $a \leftarrow t[1] - t[0] - 1$ 
3    $b \leftarrow n + t[0] - t[1] - 1$ 
4    $d_1 \leftarrow \min(a, b)$ 
5
6    $a \leftarrow t[2] - t[1] - 1$ 
7    $b \leftarrow n + t[1] - t[2] - 1$ 
8    $d_2 \leftarrow \min(a, b)$ 
9
10   $a \leftarrow t[2] - t[0] - 1$ 
11   $b \leftarrow n + t[0] - t[2] - 1$ 
12   $d_3 \leftarrow \min(a, b)$ 
13  return  $\max(d_1, d_2, d_3)$ 

```

de acuerdo a los tamaños de cada uno de sus triángulos. Contamos cuántos thrackles pertenecían a cada una de estas categorías dando como resultados los mostrados en la Tabla 5.2.

De estos resultados podemos observar que los triángulos de tamaño más grande tienen mayor probabilidad de aparecer en un thrackle de triángulos maximal. En especial parece ser que esto es más apreciable a medida que el valor de n aumenta.

En el capítulo anterior mencionamos cómo encontrar un orden adecuado para los triángulos puede hacer más rápida la búsqueda de un thrackle de un tamaño específico, terminando al encontrar el primer thrackle del tamaño deseado. Como un triángulo de mayor tamaño tiene mayor probabilidad de aparecer en un thrackle. Ordenamos los triángulos de forma que los de mayor tamaño sean explorados al principio, esperando reducir la cantidad de nodos explorados en el árbol de búsqueda. Reduciendo a la vez el tiempo de ejecución.

Implementamos una versión modificada del Algoritmo 2 que se detiene al encontrar el primer thrackle de un tamaño específico. Ejecutamos este algoritmo para el conjunto de 10 puntos en posición convexa, buscando th-

rackles de tamaño 12 (que sabemos es el tamaño máximo para este conjunto de puntos). Ordenando los triángulos lexicográficamente de acuerdo a las etiquetas de sus vértices, se recorrieron 2,137,586 nodos antes de encontrar el primer thrackle. Al utilizar el orden aleatorio durante 1,000 iteraciones (que fue la mejor estrategia para puntos en posición general), el algoritmo recorre 200,000 nodos. Cuando los ordenamos utilizando el tamaño de los triángulos, de mayor a menor, el algoritmo recorre 3,459 nodos, esto es el 0.16 % del número de nodos recorridos utilizando el orden lexicográfico, y el 2.7 % del número de nodos en el promedio al utilizar un orden aleatorio.

Otra forma de comparar dos triángulos sobre conjuntos de puntos en posición convexa es comparar el tamaño de sus aristas lexicográficamente, en vez de tomar sólo el tamaño de la arista más grande. Si ordenamos los triángulos de acuerdo a este criterio, el número de nodos visitados por el algoritmo se reduce a 236. Esto reduce el número de nodos a 6.8 % del número de nodos utilizando el tamaño de los triángulos (un 0.12 % del número de nodos utilizando un orden aleatorio).

Esto muestra la gran reducción de tiempo que se puede obtener al elegir el orden correcto de los triángulos antes de hacer la búsqueda. Sería interesante generalizar este criterio de orden de triángulos, para triángulos en posición general.

Hasta el momento la reducción del tiempo de ejecución sólo se presenta cuando el algoritmo se detiene en el primer thrackle. Todavía desconocemos la razón por la que el Algoritmo 2 no reduce el número de nodos recorridos al modificar el orden de los triángulos. De encontrar una versión de este algoritmo que sí reduzca el número de nodos, podría investigarse algún criterio de orden de triángulos en posición general, para calcular el valor de T_n para $n > 9$.

$n = 6$		
Tamaño		Número de thrackles
2	1	
3	1	16
2	2	12
0	4	2

$n = 7$		
Tamaño		Número de thrackles
2	1	
6	1	14
5	2	14
7	0	2

$n = 8$			
Tamaño			Número de thrackles
3	2	1	
4	4	0	96
3	5	0	24

$n = 9$			
Tamaño			Número de thrackles
3	2	1	
8	2	0	1,269
7	3	0	684
8	1	1	360
9	1	0	270
6	4	0	231
7	2	1	225
6	3	1	54
9	0	1	45
5	5	0	18

$n = 10$				
Tamaño				Número de thrackles
4	3	2	1	
5	6	1	0	22,140
5	7	0	0	10,580
5	5	2	0	7,960
4	7	1	0	3,380
4	6	2	0	2,960
4	8	0	0	320
3	7	2	0	120

Tabla 5.2: Número de thrackles maximales en el conjunto de n puntos en posición convexa, clasificados por tamaño de sus triángulos.

Capítulo 6

Conclusiones

En el trabajo de tesis presentamos varios resultados relacionados con el problema de calcular el máximo número de triángulos que puede tener un thrackle de triángulos. Estudiamos dos instancias del problema, dependiendo de la posición del conjunto de puntos sobre el que definimos un thrackle. Primero trabajamos con conjuntos de puntos en posición general, y después trabajamos el caso específico para conjuntos de puntos en posición convexa.

Para conjuntos de n puntos en posición convexa, mostramos un thrackle para todo entero n cuyo tamaño tiene crecimiento de orden $\frac{n^2}{9}$. Este ejemplo prueba la existencia de thrackles de este tamaño para todo n , y da una cota inferior del tamaño de thrackles de triángulos, presentada en el Teorema 15.

Para conjuntos de n puntos en posición general mostramos el valor del tamaño máximo de un thrackle para $3 \leq n \leq 7$. Para $8 \leq n \leq 9$ hicimos búsquedas computacionales sobre los conjuntos de puntos que representan todos los posibles tipos de orden. Como resumen, encontramos que para $6 \leq n \leq 9$, el tamaño máximo de un thrackle de triángulos es igual a la cota inferior presentada en el Teorema 15. Lo que da evidencias para creer que la Conjetura 5 es cierta, es decir, que el tamaño de un thrackle no puede ser mayor a $\frac{n^2}{9} + 2$.

Para $n = 10$, pese a que el conjunto de puntos en posición convexa tiene máximo thrackles de tamaño 12, existen configuraciones de puntos en los que el tamaño máximo de un thrackle es 11. Es decir que el tamaño máximo de un thrackle para todo conjunto de 10 puntos en posición general es 11. Este resultado para $n = 10$ es una cota inferior y no una cota superior como la que obtuvimos para valores de n mas pequeños. Por esto, queda abierta la pregunta de la existencia de un conjunto de 10 puntos que tenga un thrackle con más de 12 triángulos. Estos resultados no contradicen la Conjetura 5,

pero prueban que la cota inferior presentada en el Teorema 15, no puede extenderse para conjuntos de puntos en posición general manteniendo los mismos valores.

Desafortunadamente el tiempo de ejecución de un algoritmo que hace una búsqueda exhaustiva de todos los posibles thrackles en un conjunto de n puntos tiene un crecimiento muy rápido como función de n . Por esto, los recursos computacionales disponibles y los algoritmos presentados no fueron suficientes para hacer una búsqueda exhaustiva para $n = 10$. Es decir que no verificamos que no existe un thrackle de tamaño 13 sobre un conjunto de 10 puntos (lo que refutaría la Conjetura 5).

Pese a esto, el Algoritmo 2 que hace la búsqueda exhaustiva de thrackles podría recibir modificaciones que mejoren su tiempo de ejecución. Por un lado creemos que el árbol de búsqueda que recorre el algoritmo podría tener menos nodos. Además, podría eliminarse trabajo duplicado si se precalcula una gráfica que represente las intersecciones de todos los triángulos entre sí. El Algoritmo 1 que busca un único thrackle puede tener un mejor tiempo de ejecución al alterar el orden en el que se recorren los triángulos, el Algoritmo 2 por el contrario siempre recorre el mismo número de nodos sin importar el orden de los triángulos. Como desconocemos si esto es una característica del problema o de la implementación, cambios al Algoritmo 2 podrían permitir usar la misma estrategia de usar ordenamientos aleatorios de los triángulos para tener un mejor tiempo de ejecución. Todo esto, sumado a que la búsqueda sobre cada tipo de orden puede ser paralelizada por completo, nos hace creer que con más recursos computacionales como los disponibles con la computadora Abacus, sería posible hacer la búsqueda sobre todos los tipos de orden para $n = 10$. Con esto se obtendría más evidencia de que la Conjetura 5 es cierta, o se encontraría un contraejemplo que la refute (esta opción parece poco probable).

Otro problema en el que trabajamos fue estimar cuántos thrackles máximos tiene cada conjunto de n puntos. En las búsquedas exhaustivas que hicimos, contamos cuántos thrackles máximos tenía cada conjunto de puntos. De acá proponemos la Conjetura 6 que intenta caracterizar cuáles conjuntos de puntos tienen el mayor número de thrackles máximos. Es interesante que esta caracterización, aunque incluye al conjunto de puntos en posición convexa, no se limita sólo a este tipo de orden sino a una clase de conjuntos de puntos más amplia.

Al estudiar el problema de contar el número de thrackles específicamente para el caso de puntos en posición convexa, obtuvimos de forma analítica, una cota inferior para este valor. El resultado final se presenta en el Teorema 18 que da una expresión para el número mínimo de thrackles distintos

sobre el conjunto de puntos en posición convexa. Este resultado se basa en la construcción presentada en la demostración del Teorema 15 y conteos combinatorios del número de 1-factorizaciones y 2-factores de la gráfica bipartita completa. Este resultado también da una cota del mínimo tiempo de ejecución de cualquier algoritmo que intente enumerar todos los thrackles para conjuntos de puntos en posición convexa. Además como se cree que el conjunto de puntos en posición convexa está entre los que más thrackles maximales tiene (por la Conjetura 6) con respecto a los demás tipos de orden, creemos que este será el peor tiempo de ejecución al hacer una búsqueda exhaustiva.

Finalmente proponemos dos formas alternativas de ordenar los triángulos de un conjunto de puntos en posición convexa y mostramos que utilizándolos obtenemos una reducción significativa en los tiempos de ejecución del Algoritmo 1. Una pregunta que queda abierta es si existe una forma de ordenar los triángulos de conjuntos de puntos en posición general para obtener mejoras similares en los tiempos de ejecución de las búsquedas de thrackles sobre los demás tipos de orden.

El código fuente utilizado para realizar las búsquedas computacionales puede ser consultado en:

<https://github.com/cinvestavcs-mexicocity/Triangle-thrackles>

También se desarrolló una interfaz gráfica para visualizar conjuntos de puntos, por simplicidad, el código fuente de esta aplicación no se encuentra en el mismo repositorio. Sin embargo, el repositorio contiene información de cómo obtener una copia del código fuente de la aplicación.

Bibliografía

- [1] P. Erdős, “On sets of distances of n points,” *The american mathematical monthly*, vol. 53, no. 5, pp. 248–250, 1946.
- [2] J. L. Alvarez, J. C. Lagos, and J. Urrutia, “Crossing families and self crossing hamiltonian cycles,” vol. XVI of *Encuentros de Geometría Computacional, Barcelona, July 1-3*, 2015.
- [3] L. Lovász, J. Pach, and M. Szegedy, “On conway’s thrackle conjecture,” *Discrete & Computational Geometry*, vol. 18, pp. 369–376, Dec 1997.
- [4] B. Aronov, P. Erdős, W. Goddard, D. J. Kleitman, M. Klugerman, J. Pach, and L. J. Schulman, “Crossing families,” *Combinatorica*, vol. 14, no. 2, pp. 127–134, 1994.
- [5] P. Erdős, “On the combinatorial problems which I would most like to see solved,” *Combinatorica*, vol. 1, no. 1, pp. 25–42, 1981.
- [6] C. Huemer, D. Lara, and C. Rubio-Montiel, “Coloring decompositions of Complete Geometric Graphs: a problem inspired by the Erdős-Faber-Lovász Conjecture,” *arXiv preprint arXiv:1610.01676*, 18 de Julio 2017.
- [7] G. Chartrand and P. Zhang, *Chromatic graph theory*. CRC press, 2008.
- [8] D. E. Knuth, *The Art of Computer Programming, Volume 3: (2Nd Ed.) Sorting and Searching*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 1998.
- [9] J. Pach, “Geometric graph theory,” tech. rep., Cambridge University Press, 1999.
- [10] O. Aichholzer, F. Aurenhammer, T. Hackl, F. Hurtado, A. Pilz, P. Ramos, J. Urrutia, P. Valtr, and B. Vogtenhuber, “On k -convex point sets,” *Computational Geometry*, vol. 47, no. 8, pp. 809 – 832, 2014.

- [11] J. E. Goodman and R. Pollack, “Multidimensional sorting,” *SIAM Journal on Computing*, vol. 12, no. 3, pp. 484–507, 1983.
- [12] O. Aichholzer, F. Aurenhammer, and H. Krasser, “Enumerating order types for small point sets with applications,” *Order*, vol. 19, no. 3, pp. 265–281, 2002.
- [13] D. Romero and A. Sánchez-Arroyo, “Adding evidence to the Erdős-Faber-Lovász Conjecture,” *ARS Combinatorica*, no. 85, pp. 71–84, 2007.
- [14] D. Romero and F. Alonso-Pecina, “The Erdős–Faber–Lovász Conjecture is true for $n \leq 12$,” *Discrete Mathematics, Algorithms and Applications*, vol. 06, no. 03, p. 1450039, 2014.
- [15] G. Araujo-Pardo and A. Vázquez-Ávila, “A note on Erdős-Faber-Lovász Conjecture and edge coloring of complete graphs,” *arXiv preprint arXiv:1605.03374*, May 2016.
- [16] J. Cano, L. F. Barba, T. Sakai, and J. Urrutia, *On Edge-Disjoint Empty Triangles of Point Sets*, pp. 83–100. New York, NY: Springer New York, 2013.
- [17] E. Boros and Z. Füredi, “The number of triangles covering the center of an n -set,” *Geometriae Dedicata*, vol. 17, no. 1, pp. 69–77, 1984.
- [18] H. Hanani, “Balanced incomplete block designs and related designs,” *Discrete Mathematics*, vol. 11, no. 3, pp. 255–369, 1975.
- [19] C.-M. Fu, H.-L. Fu, and C. Rodger, “Decomposing $k_n \cup p$ into triangles,” *Discrete Mathematics*, vol. 284, no. 1, pp. 131 – 136, 2004. Special Issue in Honour of Curt Lindner on His 65th Birthday.
- [20] O. Aichholzer, F. Aurenhammer, and H. Krasser, “Database of point sets representing all order types.” Consultada por última vez en Junio de 2017 de: <http://www.ist.tugraz.at/aichholzer/research/rp/triangulations/ordertypes/>.
- [21] D. E. Knuth, *The Art of Computer Programming, Volume 4, Pre-fascicle 5b: Introduction to Backtracking*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., November 2016.