



Universidad Nacional Autónoma de México
Posgrado en Ciencia e Ingeniería de la Computación

Complejidad computacional en problemas de geometría combinatoria y computacional

T E S I S

Que para optar por el grado de:
Maestro en ciencia e ingeniería de la computación

P R E S E N T A:

Andrés Fuentes Hernández

Director de tesis: Dr. José David Flores Peñaloza

Ciudad de México, Febrero de 2022



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

AGRADECIMIENTOS

Fiando en el instinto que me empuja,
desprecio los peligros que señalas.
«El ave canta aunque la rama cruja,
como que sabe lo que son sus alas».

Salvador Díaz Mirón

A mis padres, que siempre me han apoyado e impulsado en mi formación profesional y personal. Gracias a sus sacrificios pude tener una visión más amplia del mundo, necesitaría más de una vida para expresar todo mi agradecimiento hacia ustedes.

A mis abuelas Amelia y Josefina de las cuales nunca me faltó su cariño y me enseñaron resiliencia y paciencia. En general a mi familia que siempre me ha apoyado.

A mi asesor, el Dr. José David Flores Peñaloza que me guio durante la realización de este trabajo y me mostró lo emocionante que puede ser la investigación en el área de Teoría de la Computación.

A mis sinodales, el Dr. Carlos Bruno Velarde Velázquez, el Dr. Sergio Rajsbaum Gorodesky, el Dr. Armando Castañeda Rojano y el Dr. César Hernández Cruz, por su tiempo y sus comentarios que me ayudaron a mejorar este trabajo.

Al posgrado de ciencias e ingeniería de la computación, por todo lo que aprendí gracias a todos los profesores con los que tuve la oportunidad de estudiar.

Investigación realizada gracias al Programa UNAM-PAPIIT IN120520.

ÍNDICE GENERAL

Introducción	5
1. Computabilidad y Complejidad computacional	6
1.1. Máquinas de Turing	7
1.1.1. Variantes de la máquina de Turing	8
1.2. Maquinas RAM	9
1.3. Notación Big O	9
1.4. Marco de trabajo	10
1.5. Clases de Complejidad	10
1.6. Reducciones	11
1.7. Problema SAT	11
2. Triángulos localmente sin traslape (TLST)	12
2.1. Definición y análisis del problema TLST-T	13
2.1.1. TLST-T es NP-completo	13
2.2. Definición y análisis del problema TLST-C	15
2.2.1. TLST-C es NP-completo	15

2.2.2. Estructura del conjunto de triángulos localmente sin traslape inducido por la reducción Φ	16
2.2.3. Prueba de NP-completitud	17
2.3. Análisis experimental del problema TLST	19
3. Polígono arcoíris	23
3.1. El problema del polígono arcoíris es NP-completo	24
3.1.1. Reducción 1-IN-3-SAT a polígono arcoíris	24
3.1.2. Posición general en el conjunto de puntos generado por la reducción Φ	29
3.1.3. Estructura del polígono arcoíris inducido por la reducción Φ	35
3.1.4. Prueba de NP-completitud	38
4. Árbol arcoíris	42
4.1. El problema del árbol arcoíris es NP completo	43
4.1.1. Reducción 1-IN-3-SAT al problema del árbol arcoíris	43
4.1.2. Estructura del árbol arcoíris inducido por la reducción Φ	44
4.1.3. Prueba de NP-completitud	45
4.1.4. Reducción alternativa 1-IN-3-SAT al problema del árbol arcoíris	47
4.1.5. Estructura del árbol arcoíris inducido por la reducción Γ	48
4.1.6. Prueba de NP-completitud a través de la reducción Γ	50
Conclusiones	52

ÍNDICE DE FIGURAS

1.1. Máquina de Turing	7
2.1. Ejemplos de triángulos localmente sin traslape y con traslape.	13
2.2. Ejemplo triangulación de un conjunto de puntos	14
2.3. Puntos resultantes de la reducción de 3-SAT a triángulos localmente sin traslape.	16
2.4. Triángulos resultantes de la reducción de 3-SAT a triángulos localmente sin traslape.	19
2.5. Ejemplo de un conjunto de 8 triángulos localmente sin traslape en un conjunto de 9 puntos.	22
3.1. Polígono arcoíris de 3 aristas formado en un conjunto de 8 puntos con 5 colores diferentes.	23
3.2. Artefacto de variable para la reducción 1-IN-3-SAT a polígono arcoíris.	25
3.3. Regiones de clausulas usadas en la reducción 1-IN-3-SAT a polígono arcoíris.	26
3.4. Acercamiento a los puntos de cláusula correspondientes a las ocurrencias de la $\neg X_1$ producidos por la reducción 1-IN-3-SAT a polígono arcoíris.	27
3.5. Estructura resultante de la reducción 1-IN-3-SAT a polígono arcoíris.	28

3.6. Primer análisis de regiones que delimitan los artefactos de variable.	30
3.7. Segundo análisis de regiones que delimitan los artefactos de variable.	33
3.8. Recta con pendiente máxima para evitar que sean colineales 3 puntos entre 3 regiones.	36
3.9. Análisis de los vértices del polígono arcoíris resultante de la reducción.	38
3.10. Ejemplo de línea separadora en artefacto de variable.	40
3.11. Ejemplo de polígono arcoíris producido por la reducción 1-IN-3-SAT a polígono arcoíris.	41
4.1. Árbol arcoíris de 3 segmentos formado en un conjunto de 8 puntos con 5 colores diferentes.	42
4.2. Puntos resultantes de la reducción del problema 1-IN-3-SAT a árbol arcoíris.	44
4.3. Árbol arcoíris resultante de la reducción del problema 1-IN-3-SAT a árbol arcoíris.	47
4.4. Puntos resultantes de la reducción alternativa del problema 1-IN-3-SAT a árbol arcoíris.	48
4.5. Árbol arcoíris resultante de la reducción alternativa del problema 1-IN-3-SAT a árbol arcoíris.	49

INTRODUCCIÓN

La geometría combinatoria fue descrita por Hadwiger y Debrunner [6] como un área de bases simples, cuyos problemas surgen de las relaciones primitivas entre elementos básicos de la geometría. A pesar de que el planteamiento de estos problemas es usualmente simple, de estos surge una complejidad asociada a las posibilidades combinatorias de sus elementos y las relaciones entre ellos. Por lo tanto, este trabajo se enfocará en problemas afines al área de la geometría combinatoria y computacional, explorándolos a través del punto de vista de la complejidad computacional; la cual estudia el tiempo y el espacio ocupado por un modelo de computación al decidir un lenguaje en función del tamaño de la entrada.

En el primer capítulo se define computabilidad de una manera intuitiva, posteriormente, se introduce el concepto de máquinas de Turing como modelo de computación. De igual manera, se define complejidad computacional y conceptos como notación Big O, clases de complejidad y reducciones (transferencia de complejidad). Finalmente, se introduce el problema 3-SAT y sus variantes, ya que en capítulos posteriores se muestran reducciones de variantes de este problema hacia problemas de geometría combinatoria.

Del segundo al cuarto capítulo se presentan resultados de dureza computacional originales. En el tercero se estudia la complejidad del problema de los triángulos localmente sin traslape extraído del artículo *New bounds on the maximum number of locally non-overlapping triangles in the plane* [1], en el que se realiza un estudio experimental del problema y además se estudia la complejidad de 2 variantes del mismo. Mientras que, en el siguiente capítulo se estudia la complejidad del problema del polígono arcoíris más pequeño, extraído del artículo *Rainbow polygons for colored point sets in the plane* [7]. Asimismo, del problema anterior surge la variante del árbol arcoíris mas pequeño, del cual se estudia su complejidad en el quinto capítulo.

Finalmente, se da un resumen de las reducciones encontradas y los problemas que permanecen abiertos después de este trabajo.

CAPÍTULO 1

COMPUTABILIDAD Y COMPLEJIDAD COMPUTACIONAL

A fin de otorgar una noción sobre los conceptos de computabilidad y complejidad computacional supongamos 2 agentes, un profesor encargado de hacer preguntas y un alumno encargado de responderlas. Si dicho profesor pregunta al alumno: “¿es 7919 un número primo?”, entonces, el alumno deberá seguir una serie de pasos para determinar si dicho número es primo, para finalmente contestar que efectivamente aquel número es un número primo; a este tipo de preguntas se les conoce como decidibles, porque tarde o temprano el alumno llegará a una respuesta, la cual quedará circunscrita a *sí* o *no*. Ahora bien, si el profesor le pide al alumno determinar la posición del número primo con respecto a los demás números primos, éste tras otra serie de pasos podrá dar con una respuesta concreta que irá más allá de un sí o no; a este tipo de preguntas las llamaremos computables, porque en ellas es posible llevar a cabo un cálculo que arroje una respuesta no binaria.

A razón de lo anterior, podemos definir *computabilidad* como la propiedad de una pregunta de ser respondida a través de una serie de pasos (algoritmo); por lo que una pregunta no computable será aquella que nunca podremos responder de manera correcta si se sigue un algoritmo. De este modo, si el profesor del ejemplo anterior lleva a cabo las enunciaciones: “la siguiente aseveración es falsa” y, después, “la aseveración anterior es verdadera”; entonces, el alumno se interrogará si la segunda enunciación es verdadera o falsa, una cuestión indecidible o incomputable, ya que si la segunda aseveración es cierta: la primera tendría que ser falsa, pero si ésta es falsa, la segunda es también falsa, contradiciendo el razonamiento inicial y resultando en una paradoja.

Con la intención de brindar una noción sobre el término de complejidad

computacional, nos enfocaremos en las preguntas decidibles y computables. Definiremos complejidad como el tiempo y espacio en el pizarrón que le toma al alumno responder una pregunta, por ejemplo: “¿cuántas canicas hay en una caja?”, para responder tal interrogante el alumno contará cada una de las canicas y el tiempo de respuesta está relacionado de manera lineal con del número de canicas que el profesor colocase en la caja; ahora bien, si el profesor pregunta: “¿es n un número par?” entonces el alumno podrá responder casi de manera inmediata con tan solo analizar el ultimo dígito de n . Y observemos que la cantidad de tiempo y espacio utilizado para resolver la cuestión está en función de la pregunta, específicamente en el tamaño de los datos de entrada que proporciona la pregunta. Es decir, el área de complejidad estudia la cantidad de recursos que el alumno necesita para responder a cierta familia de preguntas.

Existen distintos modelos de computación, por ejemplo: máquinas de acceso a registros (RAM), cálculo lambda, modelo Kleene-Herbrand y Godel (KHG) y máquinas de Turing [8]. Para fines de este trabajo, solo se abordarán las máquinas de Turing, ya que el modelo es suficiente para los estudios de complejidad presentados.

1.1. Máquinas de Turing

Una máquina de Turing es un modelo de computación basado en una máquina abstracta que tiene como entrada una cadena de símbolos escrita sobre una cinta infinita. Utilizando un cabezal que puede leer y escribir en la cinta, la máquina transforma la cadena de entrada en una cadena de salida a través de una serie de pasos. En cada paso, la máquina decide qué símbolo escribir sobre la cinta y, con base en ello se mueve a la izquierda o a la derecha; tal decisión estará basada en el símbolo actual que está leyendo el cabezal y, también, al estado de la máquina que dependerá de los estados anteriores y los símbolos leídos previamente. Por cierto, la máquina se detiene cuando llega a un estado final.

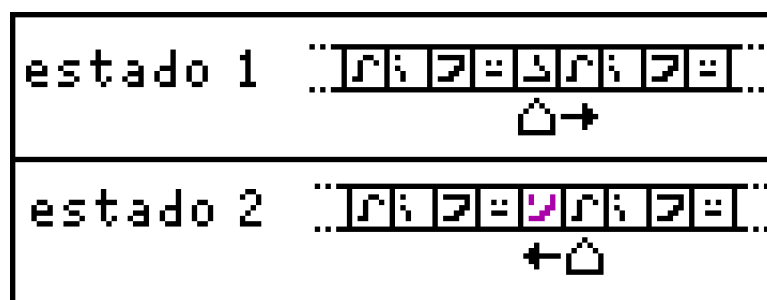


Figura 1.1: Máquina de Turing conformada por una cinta infinita de símbolos, un cabezal que puede leer y escribir en la cinta y que puede moverse a la izquierda o la derecha.

Cuando la máquina llega a un estado final, la cinta contiene el resultado del cómputo realizado, en algunas ocasiones no es importante que la máquina devuelva un cómputo sobre la cinta, solo que llegue a un estado final. Por lo que, siguiendo la analogía del profesor y el alumno, una máquina de Turing es

el equivalente del alumno y la cinta, es la pregunta realizada por el profesor. Entonces, cuando una máquina de Turing alcanza un estado final dada una cadena de entrada x , diremos que x es Turing computable.

De manera más formal una máquina de Turing puede describirse como una 7-tupla $M_T = \langle Q, \Gamma, b, \Sigma, \delta, q_0, F \rangle$.

- Q , es el conjunto de estados finito y no vacío.
- Γ , es el conjunto de símbolos del alfabeto de cinta, finito y no vacío.
- $b \in \Gamma$, es el símbolo vacío.
- $\Sigma \in \Gamma \setminus b$, es el conjunto de símbolos de entrada.
- $q_0 \in Q$, es el estado inicial.
- $F \subseteq Q$, es el subconjunto de estados finales.
- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times \{I, D\}$, es la función de transición que arroja el estado siguiente en función del estado actual y del símbolo leído actual, además indica si el movimiento es a la izquierda I o a la derecha D .

1.1.1. Variantes de la máquina de Turing

A partir de la máquina de Turing propuesta en 1936, se desarrollaron algunas variantes de las cuales se dará una breve explicación [9].

- **Máquinas de Turing deterministas:** una máquina de Turing como la definida en la sección 1.1 es determinista ya que se compone de una función de transición δ es decir tiene una sola tripleta de salida $s \in Q \times \Gamma \times \{I, D\}$ por cada par de entrada $e \in Q \times \Sigma$.
- **Máquinas de Turing no deterministas:** una máquina de Turing no determinista puede tener más de una sola tripleta de salida s por cada par de entrada e . Es decir en lugar de una función de transición esta tiene una relación de transición. Esto se puede interpretar como que dado el estado actual y el símbolo leído en la cinta, la máquina no determinista tiene múltiples caminos a elegir.
- **Máquinas de Turing multicinta:** como su nombre lo indica, las máquinas de Turing multicinta tienen más de una cinta de trabajo.
- **Máquinas de Turing multicabezal:** tienen más de un cabezal sobre la cinta de entrada.
- **Máquinas de Turing compuestas:** utilizan otras máquinas de Turing con funciones específicas, y los resultados de éstas son integrados al resultado de la máquina principal.

1.2. Maquinas RAM

Una maquina de acceso aleatorio, RAM por sus siglas en ingles, es un modelo de computación que consiste en un programa finito que trabaja con un numero infinito de registros denotados X_0, X_1, X_2, \dots . Cada registro puede contener un número entero arbitrario. Las instrucciones de una RAM consisten en operaciones de lectura/escritura sobre los registros, así como operaciones aritméticas como suma y resta sobre el contenido de los mismos. También cuenta con operaciones de ramificación que permiten modificar el flujo de ejecución del programa. El modelo RAM y las maquinas de Turing son equivalentes [11], es decir una maquina de Turing puede simular al modelo RAM con un costo polinomial en tiempo y viceversa. Durante este trabajo se utilizará al modelo RAM para el estudio de complejidad de los algoritmos.

1.3. Notación Big O

La notación Big O nos dice que tan rápido crece una función y es usada para medir la complejidad de un algoritmo; por ejemplo, si una máquina de Turing tiene como entrada una cadena de n símbolos que debe contar, entonces es fácil ver que en n pasos la tarea será completada entonces diremos que el tiempo de ejecución del programa es $T(n) = n$. Observemos que $T(n)$ crece linealmente de acuerdo al tamaño de la entrada o que el algoritmo tiene complejidad $O(n)$ en tiempo. Otro ejemplo es, si un algoritmo toma un número de pasos $T(n) = 1000 * n^2 + n$, diremos que el algoritmo tiene complejidad $O(n^2)$ en tiempo [3].

Algunas definiciones formales de complejidad son:

- $f(n) \in O(g(n))$: Notación Big O, $|f|$ está delimitado por arriba por g asintoticamente. Se define formalmente como:
 $\exists c \in \mathbb{R}^+, \exists n_0 \in \mathbb{N}$ tales que $\forall n \in \mathbb{N}$, si $n > n_0$ entonces $|f(n)| \leq c \cdot g(n)$.
- $f(n) = \Theta(g(n))$: Notación Big Theta, $|f|$ está delimitado por arriba y por debajo asintoticamente. Se define formalmente como:
 $\exists c_1, c_2 \in \mathbb{R}^+, \exists n_0 \in \mathbb{N}, \forall n \in \mathbb{N}$, si $n > n_0$ entonces $c_1 \cdot g(n) \leq |f(n)| \leq c_2 \cdot g(n)$.
- $f(n) = \Omega(g(n))$: Notación Big Omega, $|f|$ está delimitado por debajo asintoticamente. Se define formalmente como:
 $\exists c \in \mathbb{R}^+, \exists n_0 \in \mathbb{N}, \forall n \in \mathbb{N}$, si $n > n_0$ entonces $|f(n)| \geq c \cdot g(n)$.

1.4. Marco de trabajo

Para analizar la complejidad de problemas de decisión es necesario que utilicemos un marco formal. Partiremos de la teoría de lenguajes formales. En este marco un alfabeto Σ es un conjunto finito de símbolos y L es un lenguaje en Σ , lo que significa que L es un conjunto de cadenas formadas con símbolos de Σ . Se denota a la cadena vacía como ϵ y al lenguaje vacío como θ . También, se denota Σ^* como el conjunto de todas las cadenas que se pueden formar con los símbolos de un alfabeto Σ , incluyendo a la cadena vacía. En la teoría de lenguajes formales un problema de decisión Q es un subconjunto de Σ^* , donde $\Sigma = \{0, 1\}$, como los problemas de decisión están caracterizados por sus instancias sí, entonces podemos definir Q como un lenguaje L en $\Sigma = \{0, 1\}$ tal que:

$$L = \{x \in \Sigma^* | Q(x) = 1\}$$

Bajo este marco, diremos que un algoritmo A acepta una cadena $x = \{0, 1\}^*$, si dada la entrada x , $A(x) = 1$, de la misma manera diremos que un algoritmo A rechaza una cadena $x = \{0, 1\}^*$, si dada la entrada x , $A(x) = 0$. El lenguaje aceptado por el algoritmo A es el conjunto de cadenas $L = \{x \in \{0, 1\}^* | A(x) = 1\}$. Con base en la definición de aceptación de un lenguaje, un lenguaje L es decidido por A si toda cadena perteneciente a L es aceptada y toda cadena que no pertenezca a L es rechazada [5].

1.5. Clases de Complejidad

Se puede definir a una clase de complejidad como al conjunto de lenguajes L cuya pertenencia es determinada usando una *medida de complejidad*, por ejemplo tiempo de ejecución o cantidad de memoria usada de un algoritmo A , que determina si una cadena x pertenece a L .

Definición 1.5.1. Clase NP: Un lenguaje L esta en NP si existe una función polinomial $p : \mathbb{N} \rightarrow \mathbb{N}$ y una maquina de turing M , llamada el verificador de L , de tal manera que para toda cadena $x = \{0, 1\}^*$:

$$x \in L \Leftrightarrow \exists u \in \{0, 1\}^{p(|x|)} \text{ tal que } M(x, u) = 1$$

Si esto se satisface entonces diremos que u es un certificado para x con respecto al lenguaje L y la maquina de Turing M [3].

Definición 1.5.2. Clase P: Un lenguaje L está en P si existe un algoritmo A que decide L en tiempo polinomial [3].

1.6. Reducciones

Un lenguaje $L \subseteq \{0,1\}^*$ es *Karp reducible* en tiempo polinomial a un lenguaje $L' \subseteq \{0,1\}^*$, representado como: $L \leq_p L'$, si existe una función computable en tiempo polinomial $f : \{0,1\}^* \rightarrow \{0,1\}^*$ tal que por cada cadena $x \in \{0,1\}^*$, $x \in L$ si y solo si $f(x) \in L'$.

Definición 1.6.1. NP-duro: diremos que L' es NP-duro si por cada $L \in NP$, $L \leq_p L'$ [3].

Definición 1.6.2. NP-completo: diremos que L' es NP-completo si L' es NP-duro y $L' \in NP$ [3].

1.7. Problema SAT

El problema de satisfacibilidad booleana SAT es el primero de su clase que se comprobó como NP-completo [4], y a partir de éste se han encontrado más problemas NP-completos mediante reducciones. Tal problema consiste en encontrar una asignación de valores a las variables que vuelva verdadera a una fórmula booleana.

Existen variantes de este problema que también son NP-completas, como:

- **3-SAT:** es una variante de SAT sobre funciones booleanas en forma normal conjuntiva (CNF) es decir cláusulas de 3 literales unidas por el operador or \vee y las cláusulas a su vez unidas por el operador and \wedge .
Ejemplo: $F = (X_0 \vee X_1 \vee \neg X_2) \wedge (X_1 \vee X_2 \vee \neg X_3) \wedge (X_0 \vee \neg X_1 \vee X_3)$.
- **1-IN-3-SAT:** es una variante de 3-SAT que busca asignaciones a las variables, de tal manera que solo una literal satisfaga cada cláusula.
- **NAE3SAT:** es una variante de 3-SAT que busca que cada cláusula se satisfaga por a lo más 2 literales.

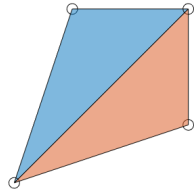
CAPÍTULO 2

TRIÁNGULOS LOCALMENTE SIN TRASLAPE (TLST)

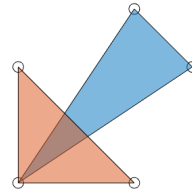
En el artículo *New bounds on the maximum number of locally non-overlapping triangles in the plane* [1] se estudia el siguiente problema: dado un entero n , ¿cuál es el máximo número de triángulos localmente sin traslape que pueden existir en todo conjunto de n puntos en posición general (sin que tres de estos estén sobre una misma recta)? Siendo que, un conjunto de triángulos localmente sin traslape, es aquél que respeta las siguientes restricciones (ver Figura 2.1):

1. Cada par de triángulos en el conjunto puede compartir a lo más un vértice.
2. Si un par de triángulos comparte un vértice, entonces no deben compartir ningún otro punto interior; si no tienen ningún vértice en común entonces pueden compartir puntos en su interior.

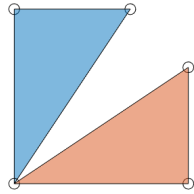
En el artículo antes referido se establecieron cotas superiores e inferiores al máximo número de triángulos localmente sin traslape que hay en un conjunto de n puntos en posición general. Pero la versión computacional del problema está abierta: dado un conjunto S de n puntos en el plano en posición general, calcular eficientemente el máximo número de triángulos localmente sin traslape que tienen vértices en S . Se conjetura que el problema de decisión asociado es NP-Completo, y a continuación se presentan generalizaciones del problema TSLT que demostramos ser NP-Completo.



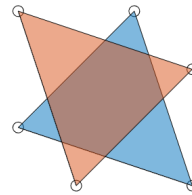
(a) Triángulos que incumplen la restricción 1, no son triángulos localmente sin traslape.



(b) Triángulos que incumplen la restricción 2, no son triángulos localmente sin traslape.



(c) Triángulos localmente sin traslape que solamente comparten un vértice.



(d) Triángulos localmente sin traslape que al no compartir vértices, pueden compartir más puntos en su interior.

Figura 2.1: Ejemplos de triángulos localmente sin traslape y con traslape.

2.1. Definición y análisis del problema TLST-T

El problema TLST tiene como entrada un entero k y un conjunto de puntos S en posición general de tamaño n , el cual induce un conjunto de triángulos de tamaño $\binom{n}{3}$ y, se debe determinar si existen k de ellos que son localmente sin traslape. Debido a la naturaleza combinatoria del problema TLST, se definió un problema más general al cual se le denominó TLST-T.

El enunciado del problema TLST-T es: dado un conjunto T de n triángulos en el plano ¿existe un conjunto $T_0 \subset T$ de cardinalidad k , cuyos elementos sean triángulos localmente sin traslape?

2.1.1. TLST-T es NP-completo

Lema 2.1.1. *El problema TLST-T está en NP*

Demostración. Un certificado del problema TLST-T es un conjunto de k triángulos, en dónde se puede verificar en tiempo $O(k^2)$ que el certificado

pertenece al lenguaje TLST-T. Esto se hace analizando cada par de triángulos en el certificado y comprobando que respeten las restricciones necesarias para triángulos localmente sin traslape. Para comprobar que cada par de triángulos no comparte ningún vértice se toma cada uno de los vértices de un triángulo y se compara contra los 3 vértices del otro, resultando en 9 comparaciones en total. Finalmente, para comprobar que cada par de triángulos no se traslapa con otro, basta con verificar que cada vértice de un triángulo no está en el interior del otro triángulo, tal comprobación es de tiempo constante por vértice. \square

Para probar la NP-dureza del problema TLST-T se redujo el problema del *Conjunto independiente de triángulos* [12] a TLST-T utilizando como reducción la función identidad. Un *conjunto independiente de triángulos* es aquel en el cual cada par de triángulos del conjunto comparte a lo más un vértice. En [12] se demuestra que *Conjunto independiente de triángulos* es NP-Completo y su enunciado es el siguiente: dada una triangulación de un polígono T , ¿existe un conjunto $T_0 \subset T$ de cardinalidad m , tal que T_0 es un conjunto independiente de triángulos? (ver Figura 2.2). Una triangulación de un polígono, es una descomposición del polígono en un conjunto de triángulos que utilizan solamente vértices del polígono y no comparten puntos en su interior entre ellos.

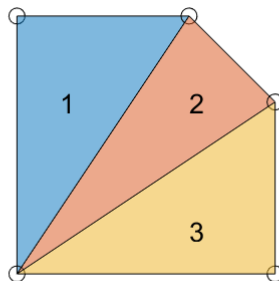


Figura 2.2: Triangulación $T = \{t_1, t_2, t_3\}$ donde $T_0 = \{t_1, t_3\}$ y $T_1 = \{t_2\}$ son ejemplos de subconjuntos independientes de triángulos.

Lema 2.1.2. *Sea T el conjunto de triángulos que forma parte de una triangulación arbitraria, si existe un conjunto independiente de triángulos $T_0 \subset T$ entonces T_0 es también un conjunto de triángulos localmente sin traslape.*

Demostración. Debido a que son parte de una triangulación, cada par de triángulos en T_0 comparte a lo más 2 vértices y ningún par comparte puntos

en su interior (no se traslapan). Por lo anterior, cumplen con la restricción 2 de los triángulos localmente sin traslape y por definición de conjunto independiente de triángulos, cada par de triángulos comparte a lo más 1 vértice. Por lo que T_0 cumple con la restricción 1 de los triángulos localmente sin traslape, entonces, T_0 es un conjunto de triángulos localmente sin traslape. \square

Lema 2.1.3. *Sea T el conjunto de triángulos que forma parte de una triangulación arbitraria, si existe un conjunto de triángulos localmente sin traslape $T_0 \subset T$ entonces T_0 es también un conjunto independiente de triángulos.*

Demostración. Si T_0 es un conjunto de triángulos localmente sin traslape, entonces por la restricción 1 solo comparten a lo más un vértice, por lo cual T_0 es un conjunto independiente de triángulos. \square

Corolario 2.1.3.1. *Por los lemas 2.1.2 y 2.1.3 se concluye que TLST-T es NP-completo.*

2.2. Definición y análisis del problema TLST-C

Dado un conjunto S de p puntos, en posición general, coloreados con k colores, un entero q y una lista Z de tripletas de colores permitidos de tamaño r . Utilizando los puntos en S como vértices: ¿existe un conjunto T , de triángulos localmente sin traslape con vértices de colores permitidos, de cardinalidad q ?

2.2.1. TLST-C es NP-completo

Consideremos la siguiente reducción Φ que mapea instancias 3-SAT a instancias TLST-C. Dada una instancia F de 3-SAT, donde F es una expresión booleana con n variables $\{X_0, X_1, X_2, \dots, X_{n-1}\}$ y m cláusulas $\{C_0, C_1, C_2, \dots, C_{m-1}\}$. Se procederá a transformar a F a un conjunto de puntos en posición convexa, de la siguiente manera: cada variable X_i se transforma en 4 puntos, sobre la curva parabólica $y = x^2$, con abscisas: $(-4i - 1), (-4i - 2), (-4i - 3), (-4i - 4)$ y colores X_i, X_{it}, X_{if}, X_i respectivamente. De igual manera, a cada cláusula C_j se le transforma en 2 puntos a los cuales se les posicionará de manera convexa sobre la misma curva que a las variables, los puntos tendrán abscisas: $(2j + 1), (2j + 2)$ y ambos serán de color C_j . Claramente los puntos agregados están en posición general y la construcción de este conjunto se realiza en tiempo $\Theta(n + m)$, (ver Figura 2.3).

Para armar la lista de tripletas de colores permitidos, por cada variable se aceptarán tripletas de la forma (X_i, X_i, X_{it}) y (X_i, X_i, X_{if}) . Por cada cláusula de

la forma $C_j = \ell_{j1} \vee \ell_{j2} \vee \ell_{j3}$ se agregarán las tripletas (C_j, C_j, ℓ_{j1}) , (C_j, C_j, ℓ_{j2}) , (C_j, C_j, ℓ_{j3}) . La lista de tripletas de colores permitidos es posible construirla en tiempo $\Theta(m + n)$, un ejemplo de la reducción de una instancia 3-SAT a TSLT-C puede observarse en la Figura 2.4.

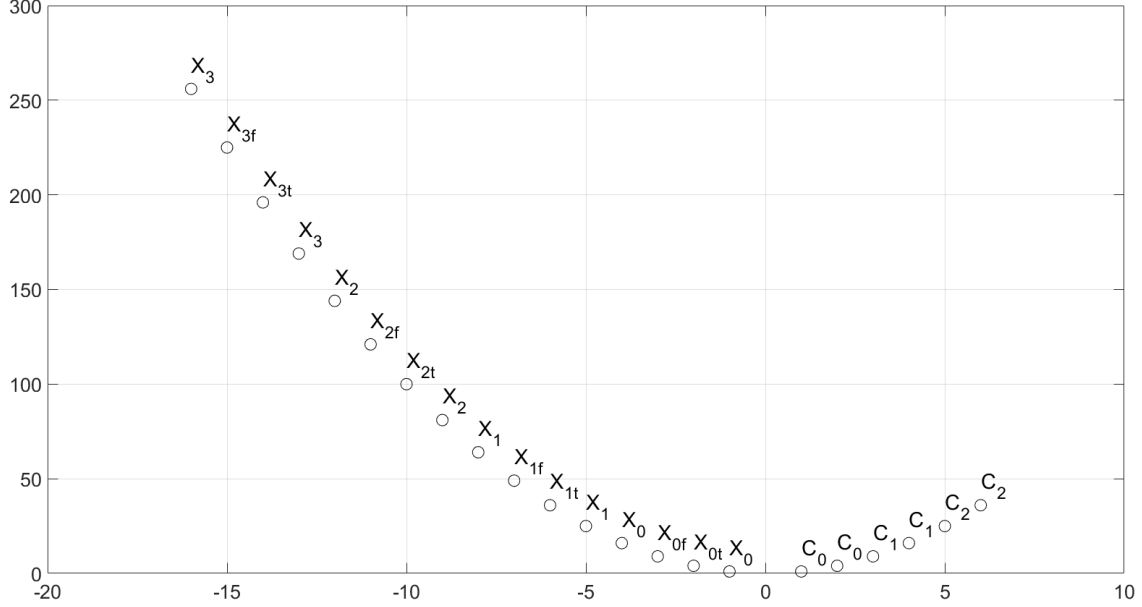


Figura 2.3: Puntos resultantes de la reducción $\Phi(F)$, con $F = (X_0 \vee X_1 \vee \neg X_2) \wedge (X_1 \vee X_2 \vee \neg X_3) \wedge (X_0 \vee \neg X_1 \vee X_3)$. La lista de colores permitidos generada es: (X_0, X_0, X_{0t}) , (X_0, X_0, X_{0f}) , (X_1, X_1, X_{1t}) , (X_1, X_1, X_{1f}) , (X_2, X_2, X_{2t}) , (X_2, X_2, X_{2f}) , (X_3, X_3, X_{3t}) , (X_3, X_3, X_{3f}) , (X_0, X_0, X_{1t}) , (C_0, C_0, X_{0t}) , (C_0, C_0, X_{1t}) , (C_0, C_0, X_{2f}) , (C_1, C_1, X_{1t}) , (C_1, C_1, X_{2t}) , (C_1, C_1, X_{3f}) , (C_2, C_2, X_{0t}) , (C_2, C_2, X_{1f}) , (C_2, C_2, X_{3t}) .

2.2.2. Estructura del conjunto de triángulos localmente sin traslape inducido por la reducción Φ

Observación 2.2.1. *En una instancia de TLST-C producida por la reducción Φ , solo se puede formar 1 triángulo por variable.*

Demostración. La reducción Φ , genera 4 puntos en posición convexa y 2 coloraciones permitidas por variable, si se forman los triángulos permitidos por las coloraciones se observa que estos comparten la arista formada por los puntos de color X_i y por definición para ser un triángulo localmente sin traslape no pueden compartir más de un vértice, por lo que solo es posible formar un triángulo por variable. \square

Observación 2.2.2. *En una instancia de TLST-C producida por la reducción Φ , se puede formar 1 o ningún triángulo por cláusula.*

Demostración. La reducción Φ , genera 2 puntos y 3 coloraciones permitidas por cláusula, si se forman los triángulos permitidos por las coloraciones se observa que estos comparten la arista formada por los puntos de color C_j , al igual que en la observación 2.2.1, esto nos lleva a que a lo más se puede formar un triángulo por cláusula. \square

2.2.3. Prueba de NP-completitud

Lema 2.2.1. *El problema TLST-C está en NP*

Demostración. Dado un certificado del problema TLST-C, éste es un conjunto de triángulos de tamaño q en el que se puede verificar en tiempo $O(q * r)$ que las coloraciones en los triángulos son permitidas y, finalmente, en tiempo $O(q^2)$ se puede verificar que sean triángulos localmente sin traslape. Entonces puede verificarse que el certificado pertenece al lenguaje TLST-C en tiempo $O(q * r + q^2)$. \square

Lema 2.2.2. *La reducción Φ es de tiempo polinomial en el tamaño de una instancia F del problema 3-SAT.*

Demostración. Como se observó en la sección 2.2.1 la creación del conjunto de puntos de variable y de cláusula toma tiempo $\Theta(n + m)$ y también la creación de la lista de tripletas de colores permitidos toma tiempo $\Theta(n + m)$. La complejidad en tiempo de la reducción Φ es $\Theta(n + m)$. \square

Lema 2.2.3. *Dada una instancia F del problema 3-SAT, con n variables, m cláusulas y una asignación de variables que satisface a F , entonces existen $n + m$ triángulos localmente sin traslape y con colores permitidos en la instancia TLST-C producida por la reducción $\Phi(F)$.*

Demostración. La reducción Φ nos provee de 2 triángulos posibles para cada variable, por lo que si la variable X_i tiene como asignación el valor verdadero, entonces tomaremos el triángulo con vértices de color (X_i, X_i, X_{if}) y dejaremos disponible el punto con color X_{it} , después se procede de manera análoga para el caso cuando la variable toma el valor falso. Si hacemos esto con todas las variables habremos formado n triángulos y, por cada cláusula C_j la reducción Φ genera 3 coloraciones permitidas y, por lo tanto, 3 triángulos posibles. Por la hipótesis sabemos que la cláusula C_j se satisface; sea una literal ℓ_{ji} que la satisface entonces incluyamos en el conjunto de triángulos al único triángulo con colores (C_j, C_j, ℓ_{ji}) , si hacemos esto con todas las cláusulas entonces generaremos m triángulos, teniendo así un conjunto T de cardinalidad $n + m$ como se quería.

Los triángulos del conjunto T son localmente sin traslape. Para probar que los triángulos son localmente sin traslape se analizarán los triángulos incidentes a cada tipo de punto. Para los puntos con colores X_i y C_j solo pueden ser parte de un solo triángulo por las observaciones 2.2.1 y 2.2.2, entre tanto los puntos de color X_{it} y X_{if} , pueden ser parte de un solo triángulo de variable, como lo muestra la observación 2.2.1, o pueden ser parte de varias cláusulas; por construcción las parejas de puntos de cláusula son consecutivas sobre la parábola, por lo tanto en aquellos triángulos con un vértice en común, se puede observar que un triángulo termina antes de que el siguiente comience. Finalmente, se concluye que los triángulos en el conjunto T son localmente sin traslape. \square

Lema 2.2.4. *Dada una instancia F del problema 3-SAT, con n variables y m cláusulas. Si existe un conjunto T de cardinalidad $n + m$ de triángulos localmente sin traslape, generados usando los puntos y la lista de colores permitidos producidos por la reducción $\Phi(F)$, entonces F es satisfacible.*

Demostración. Sabemos por las observaciones 2.2.1 y 2.2.2, que el conjunto de triángulos tiene exactamente un triángulo por cada pareja de puntos de color X_i y exactamente un triángulo por cada pareja de color C_j . Por lo cual, se construye la asignación de valores a las variables de la siguiente manera: por cada triángulo de variable si éste contiene al punto de color X_{if} , entonces la variable X_i tiene valor

verdadero y viceversa. Observemos que la pareja de puntos C_j solo puede formar triángulos con las 3 literales que la hacen cierta, sea C_j una cláusula arbitraria y sea ℓ_{ji} el color del punto que está formando un triángulo con la pareja C_j entonces ℓ_{ji} es una literal de la cláusula C_j . Destaca que la asignación de valores antes descrita hace que ℓ_{ji} se evalúe a verdadero con lo cual C_j se satisface. \square

Corolario 2.2.4.1. *Por los lemas 2.2.1 , 2.2.2, 2.2.3 y 2.2.4 se concluye que el problema TLST-C es NP-completo.*

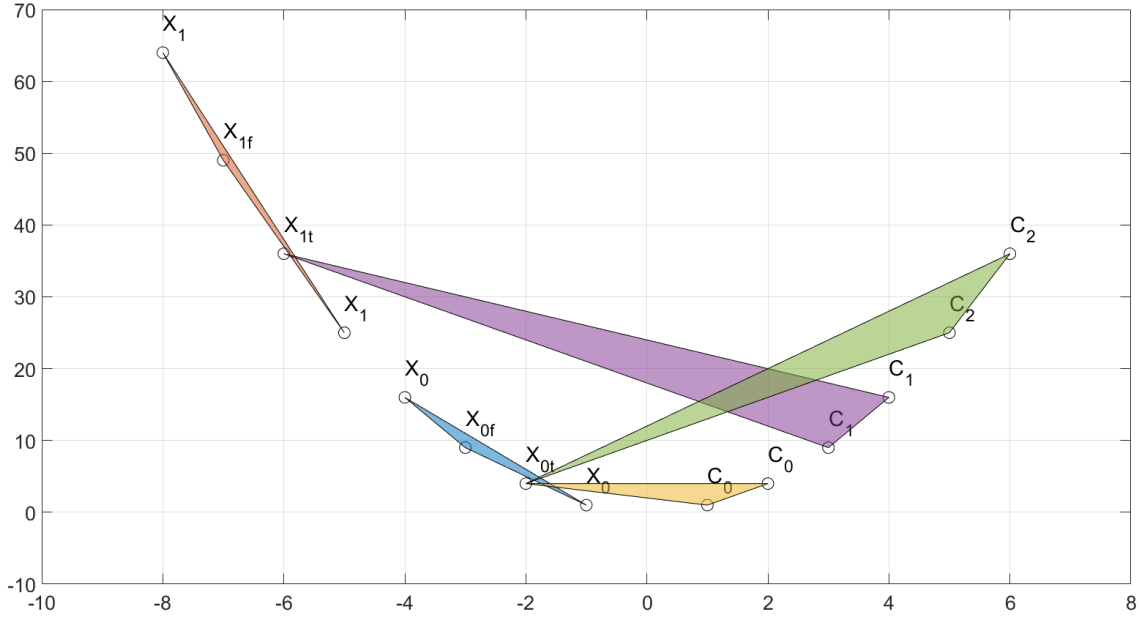


Figura 2.4: Triángulos localmente sin traslape encontrados en la reducción $\Phi(F)$, con $F = (X_0 \vee X_1 \vee \neg X_2) \wedge (X_1 \vee X_2 \vee \neg X_3) \wedge (X_0 \vee \neg X_1 \vee X_3)$. En la figura se omiten los puntos correspondientes a las variables X_2 y X_3 ya que F es satisficible si X_0 y X_1 son verdaderos.

2.3. Análisis experimental del problema TLST

Debido a que no se determinó la complejidad del problema TLST se procedió a realizar un análisis experimental. A pesar de no tener consecuencias directas para el presente trabajo, consideramos que puede ser valioso para el desarrollo de trabajos futuros. Para el análisis experimental se utilizó el concepto de tipos de orden del conjunto de puntos (clases de equivalencia de los puntos de acuerdo con su posición relativa, considerada entre todas las tripletas de puntos) [10]. Para utilizar estas clases de equivalencia se debe probar que los conjuntos de triángulos localmente sin traslape dependen solamente de los tipos de orden.

Lema 2.3.1. *Sea P un tipo de orden de n puntos. Si existe un conjunto T de triángulos*

localmente sin traslape, de cardinalidad m , usando como vértices los puntos en P , entonces T existe en todos los conjuntos de n puntos cuyo tipo de orden es P .

Demostración. Sea S_i un conjunto de puntos perteneciente al orden P , además sea T un conjunto de triángulos localmente sin traslape usando los puntos de S_i como vértices. Tomemos un par arbitrario de triángulos $t_1, t_2 \in T$. Si t_1 y t_2 comparten un vértice, digamos $v_a \in S_i$. Si ordenamos los vértices de t_1 y t_2 de acuerdo a su pendiente con v_a , veremos que los vértices de t_1 ocurren antes o después de t_2 en el orden relativo de v_a . Luego si aplicamos una transformación afín al conjunto de puntos S_i generaremos un nuevo conjunto de puntos S_f el cual también pertenece al orden P . Obsérvese que esta transformación no afecta los ordenes angulares entre los puntos por lo que para t_1 y t_2 solo su orden relativo con v_a es suficiente para mantener la propiedad de triángulos localmente sin traslape y la distancia entre los vértices no afecta. Por lo que se procede de manera similar si t_1 y t_2 no comparten vértices. \square

Para el estudio experimental, por medio de cálculos exhaustivos se encontraron los conjuntos máximos de triángulos localmente sin traslape en tipos de orden de cardinalidad 3, 4, 5, 6, 7, 8 y 9, obtenidos de la base de datos presentada en [2]. A continuación se incluye el algoritmo utilizado para encontrar un conjunto de triángulos localmente sin traslape en una nube de puntos. El algoritmo recibe un conjunto de puntos y encuentra todos los triángulos posibles en el conjunto de entrada, posteriormente crea una gráfica de conflictos, en donde un nodo representa un triángulo y una arista entre dos nodos representa dos triángulos que violan las reglas de un conjunto de triángulos localmente sin traslape. Después la gráfica generada es pasada a una función `CONJUNTOINDEPENDIENTEMAXIMO` que como su nombre lo indica encuentra un conjunto independiente máximo y regresa un subconjunto de nodos los cuales son pasados a la función `EXTRAERTRIANGULOS` que regresa el conjunto máximo de triángulos localmente sin traslape. Los resultados de este estudio se muestran en el Cuadro 4.1.

Algorithm 2.3.1 Encontrar un conjunto de triángulos localmente sin traslape.

$S \leftarrow$ conjunto de puntos en posición general

$m \leftarrow$ entero

procedure ENCONTRARTLST(m, S)

$G \leftarrow$ gráfica de conflictos G

$T \leftarrow$ Conjunto de los $\binom{n}{3}$ triángulos posibles en S

for $t_i \in T$ **do**

$G \leftarrow$ agregar nodo v_i

end for

for $t_i \in T$ **do**

for $t_j \in T$ **do**

if t_i y t_j no son triángulos localmente sin traslape **then**

$G \leftarrow$ agregar arista e_{ij} entre nodos v_i y v_j

end if

end for

end for

$G_0 \leftarrow$ CONJUNTOINDEPENDIENTEMAXIMO(m, G)

$T_0 \leftarrow$ EXTRAERTRIANGULOS(T, G_0)

return T_0

end procedure

Resultados experimentales			
n	Número de tipos de orden	Mínimo de máximos	Máximo de máximos
3	1	1	1
4	2	1	1
5	3	2	2
6	16	3	4
7	135	3	5
8	3,315	4	6
9	158,817	5	8

Cuadro 2.1: Mínimo de máximos y máximo de máximos de triángulos localmente sin traslape encontrados en los tipos de orden del 3 al 9.

A partir de los experimentos con los datos, se tiene como conjetura que el número mínimo de triángulos localmente sin traslape se da cuando todos los puntos están en el cierre convexo y que el número máximo de triángulos tiene cabida cuando las capas convexas son de tamaño mínimo. Tal hipótesis podrá ser explorada por trabajos futuros.

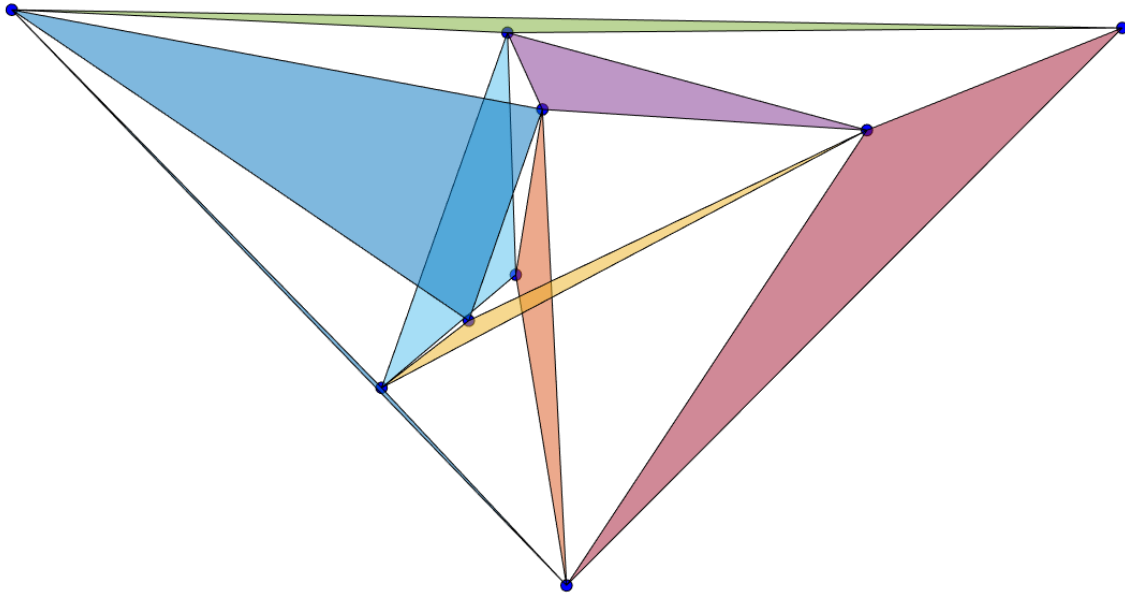


Figura 2.5: Ejemplo de un conjunto de 8 triángulos localmente sin traslape en un conjunto de 9 puntos.

En el artículo [7] se estudia el siguiente problema: dado un conjunto S de p puntos coloreados con r colores, en posición general (sin que tres de ellos estén sobre una misma recta), y un entero q : ¿es posible formar un polígono arcoíris de q aristas? Un polígono arcoíris es un polígono simple que contiene exactamente un punto de cada uno de los r colores, ya sea en su interior o en sus aristas (ver Figura 3.1).

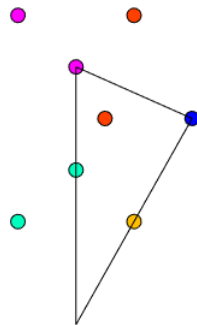


Figura 3.1: Polígono arcoíris de 3 aristas formado en un conjunto de 8 puntos con 5 colores diferentes.

3.1. El problema del polígono arcoíris es NP-completo

Para mostrar que el problema del polígono arcoíris es NP-completo se utilizará una reducción que construirá un conjunto de puntos en posición general. La reducción será desde el problema 1-IN-3-SAT, entonces se reducirá una fórmula booleana F en forma normal conjuntiva de n variables X_0, X_1, \dots, X_{n-1} y m cláusulas C_0, C_1, \dots, C_{m-1} a un polígono arcoíris de $2n + 2$ aristas en un conjunto S de $8(2n + 2) + 3m$ puntos en posición general pintados con $4(2n + 2) + m$ colores, de tal manera que $4(2n + 2) + m$ puntos de colores distintos queden en el interior del polígono arcoíris. A continuación presentaremos una distribución especial de puntos, a la que llamaremos *artefacto*, clave para la construcción del polígono arcoíris. El artefacto básico se forma con 8 puntos, separados en 4 pares, cada par pintado con un color distinto a los demás. Los puntos se colocan de tal manera que existan solo 2 tipos de recta que los separe dejando un punto de cada color en el interior de cada semiplano determinado por la recta separadora del caso. (ver Figura 3.2) Como veremos, se crearán $2n + 2$ artefactos y cada uno determinará una arista del polígono en cuestión (a cada variable se le asociarán dos artefactos y los dos artefactos extra servirán para construir dos aristas auxiliares).

Además, n de los artefactos deberán tener 2 regiones respectivamente contenidas en el interior de los semiplanos definidos por la recta separadora. Estas regiones servirán para alojar puntos correspondientes a las cláusulas. Finalmente, a cada uno de los $2n + 2$ se le aplicarán dos transformaciones afines y una translación sin modificar sus propiedades locales, con el fin de ensamblar una construcción en la que se fuerce a que todo polígono arcoíris de $2n + 2$ aristas tenga contenida cada arista en una recta separadora de un respectivo artefacto.

3.1.1. Reducción 1-IN-3-SAT a polígono arcoíris

Observación 3.1.1. *Dado un conjunto de puntos coloreados S en posición general, si en dicho conjunto existen exactamente 2 puntos de color w , entonces existen polígonos arcoíris que intersecan al segmento formado por estos 2 puntos un número impar de veces.*

Observación 3.1.2. *Dado un conjunto de puntos coloreados S en posición general, si en dicho conjunto existen exactamente 2 puntos de color w , entonces existen polígonos arcoíris en los cuales un punto de color w reside dentro de una arista del polígono mientras que el punto restante reside en la prolongación de dicha arista sin ser parte del polígono arcoíris.*

Consideremos la siguiente reducción Φ de instancias 1-IN-3-SAT a instancias del polígono arcoíris. Se supondrá una función booleana F como la anteriormente descrita. El artefacto base se construye utilizando coordenadas enteras para posicionar apropiadamente sus 8 puntos, después se le aplica transformaciones afines y, finalmente, colocarlo sobre una curva que nos garantice que una vez puesta la totalidad de artefactos todos los puntos queden en posición general. Por cada variable X_i se crearán 2 artefactos llamados H_i y V_i , el primero generara una arista horizontal mientras que el segundo dará una arista vertical; para crear un artefacto H_i se colocarán 2 puntos de color H_i^a en las coordenadas $(-60, -20)$ y $(-20, 60)$, más 2 puntos de color H_i^b en las coordenadas $(20, -60)$ y $(60, 20)$, 2 puntos de color H_i^c en las coordenadas $(1, 2)$ y $(-1, -2)$ y finalmente, 2 puntos de color H_i^d en las coordenadas $(-53, -15)$ y $(19, -51)$ (ver Figura 3.2). Conviene señalar que, la construcción de un artefacto V_i es igual al de un artefacto H_i y que además los puntos de cada artefacto están en posición general.

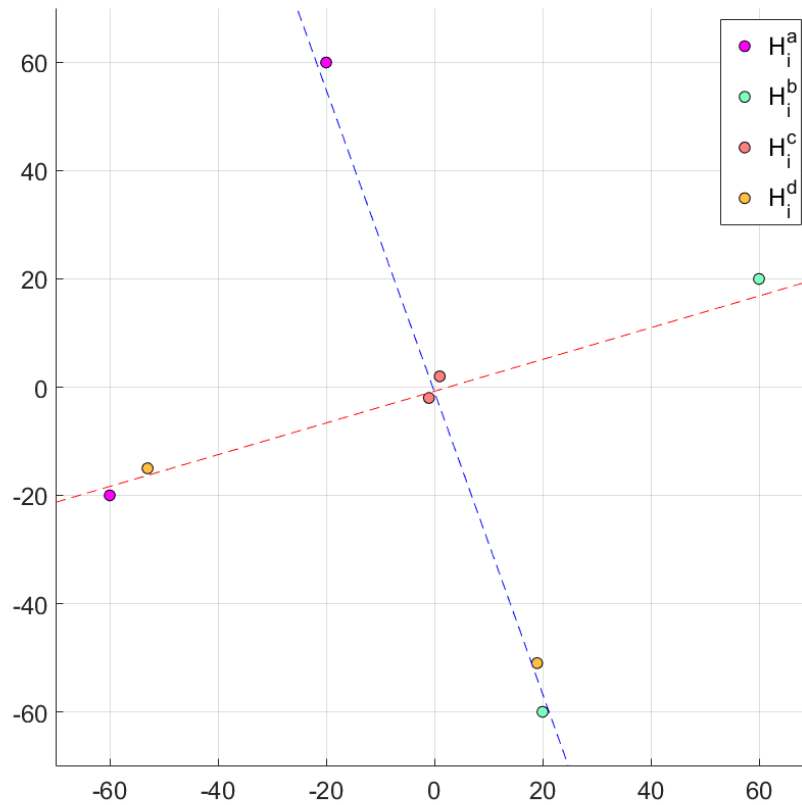


Figura 3.2: Puntos base para un artefacto de variable y ejemplo de 2 tipos de línea separadora.

Observemos que si se desea utilizar el artefacto anteriormente descrito para construir al polígono arcoíris de tamaño $2n+2$, una arista debe separar los puntos de cada artefacto, nótese que por la observación 3.1.1 dicha arista debe intersectar los segmentos formados por cada par de puntos del mismo color en el artefacto,

por lo que los segmentos posibles se dividen en 2, los que dejan por debajo al punto de color H_i^d izquierdo y los que dejan por debajo al punto de color H_i^d derecho. Se define la región triangular T_i^t con aristas $(-20, 14)$, $(-24, 14)$, $(-24, 15)$ en el artefacto H_i ; las aristas no forman parte del artefacto sólo definen una región. De igual manera se define la región triangular T_i^f con aristas $(20, -14)$, $(24, -14)$, $(24, -15)$ en el artefacto H_i . Estas regiones triangulares contendrán a los puntos de cláusula. Obsérvese que los tipos de segmento antes mencionados separan las regiones triangulares T_i^t y T_i^f de tal manera que que estas siempre pertenecen a semiplanos distintos. La elección de un tipo de segmento implica que una región quede por debajo del segmento y la otra por encima de este (ver Figura 3.3). Note que se si se deseara utilizar la observación 3.1.2 para separar los puntos de los artefactos H_i y V_i se necesitarían al menos 4 aristas, mientras que utilizando lo visto en la observación 3.1.1, solamente se utiliza una arista para separar cada artefacto.

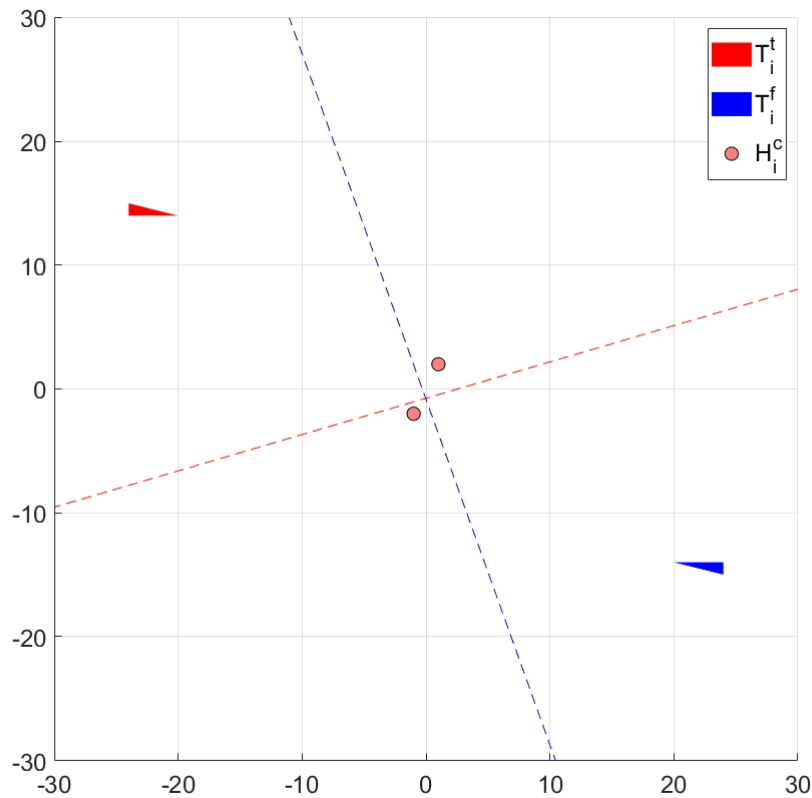


Figura 3.3: Muestra de las regiones T_i^t y T_i^f en un artefacto.

Dentro de las regiones T_i^t y T_i^f se posicionarán puntos correspondientes a aquellas cláusulas en que ocurra X_i de tal manera que la unión de ellos y los puntos ya colocados quede en posición general. Todos los puntos en cada región están en posición general con los puntos del artefacto, por lo que solo nos aseguraremos de que los puntos de cláusula dentro de estas regiones queden

en posición general entre ellos. Para ello pondremos estos puntos sobre dos arcos de parábola. Por cada cláusula de la forma $C_j = l_j^a \vee l_j^b \vee l_j^c$ que contiene a la literal X_i , se coloca un punto de color C_j en la región T_i^t en la coordenada $x_i = -20 - \frac{4j}{m-1}$ y $y_i = 14 + \frac{j^2}{(m-1)^2}$; si la cláusula contiene a la literal X_i negada se coloca el punto de color C_j en la región T_i^f en la coordenada $x_i = 20 + \frac{4j}{m-1}$ y $y_i = -\frac{j^2}{(m-1)^2} - 14$. A los artefactos V_i no se les agrega las regiones triangulares ni puntos de cláusula.

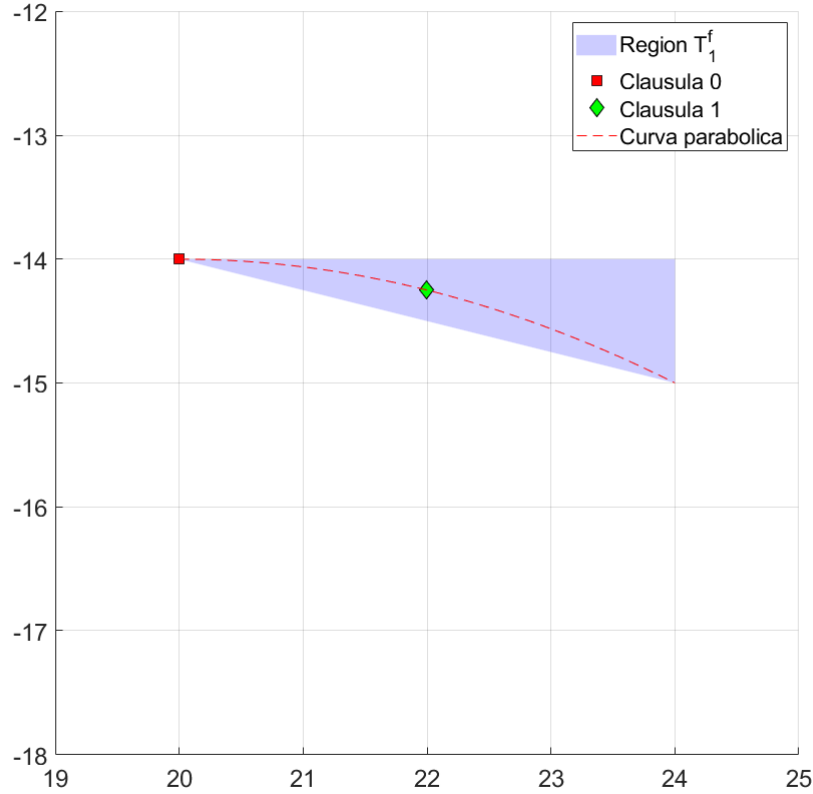


Figura 3.4: Acercamiento a los puntos de cláusula correspondientes a las ocurrencias de la $\neg X_1$ en la reducción $\Phi(F)$, con $F = (X_0 \vee \neg X_1 \vee \neg X_2) \wedge (\neg X_1 \vee X_2 \vee X_3) \wedge (X_0 \vee X_1 \vee \neg X_3)$.

Posteriormente y bajo la premisa de que las transformaciones lineales no modifican la estructura combinatoria de un conjunto de puntos, cada artefacto H_i será escalado, por un factor suficientemente pequeño, en particular $\frac{1}{640n^5}$, después la ordenada y de cada punto en el artefacto se multiplicará por un factor suficientemente pequeño en particular $\frac{1}{100*n^5*m^5}$. Finalmente, se trasladarán todos los puntos del artefacto por el vector $\langle 2i, 4i^2 \rangle$.

Asimismo, cada artefacto V_i será escalado, por un factor suficientemente pequeño, en particular $\frac{1}{640n^5}$, y después será comprimido hacia el eje vertical multiplicando por un factor suficientemente pequeño, en particular $\frac{1}{100*n^5*m^5}$. Finalmente, se trasladará todo el artefacto para que su origen quede en $(2i+1, (2i+$

1)²), sobre la parábola $y = x^2$.

Se añadirán 2 artefactos cuyo propósito es cerrar el polígono formado por los artefactos de variable. El artefacto E_h se construirá de manera similar a la realizada para un artefacto H_i , solo que no contará con puntos de cláusula en su interior y se trasladará a las coordenadas $(-\frac{3}{4}, (2n+1)^2)$. El segundo artefacto de cierre E_v se construirá de manera similar a un artefacto V_i y se trasladará a las coordenadas $(-1, 4n^2)$, (ver Figura 3.5).

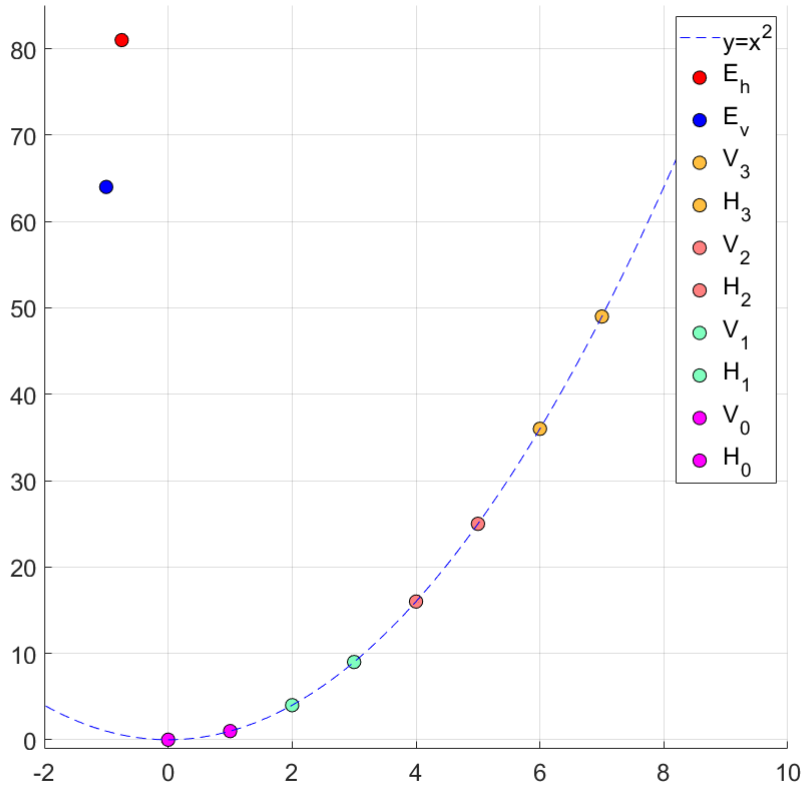


Figura 3.5: Estructura resultante de la reducción $\Phi(F)$, con $F = (X_0 \vee \neg X_1 \vee \neg X_2) \wedge (\neg X_1 \vee X_2 \vee X_3) \wedge (X_0 \vee X_1 \vee \neg X_3)$, en la imagen se aprecian los artefactos sin detalle y son representados por un punto

Observemos que la mayor pendiente de una recta que pase por dos puntos de un artefacto queda determinada por las posiciones reservadas a los puntos de colores H_i^b superior y C_{m-1} inferior, su valor es $m_h = \frac{45}{400 * n^5 * m^5}$.

De igual manera, observemos que la menor pendiente la determinan las posiciones reservadas a los puntos de colores C_0 y C_1 del triángulo T_i^t , se obtiene $m_v = \frac{200 * n^5 * m^5}{16}$.

3.1.2. Posición general en el conjunto de puntos generado por la reducción Φ

En esta sección se procederá a mostrar que el conjunto de puntos generado por la reducción Φ esta en posición general. Se utilizarán regiones cuadradas que contienen a cada artefacto, así como las pendientes máximas y mínimas posibles entre puntos dentro de estas regiones.

Lema 3.1.1. *Para todo artefacto H_j y toda recta \mathcal{L} entre un par de puntos pertenecientes a H_j se tiene que la recta \mathcal{L} pasa por encima de todos los puntos de los artefactos con un índice i con $i < j$; además, se tiene que la recta \mathcal{L} pasa por debajo de todos los puntos de los artefactos con un índice k con $j < k$.*

Demostración. Supongamos regiones cuadradas R_i que cubren los puntos de cada artefacto y están delimitadas por los puntos $(i - \Delta, i^2 + \Delta)$, $(i + \Delta, i^2 + \Delta)$, $(i + \Delta, i^2 - \Delta)$ y $(i - \Delta, i^2 - \Delta)$ donde $\Delta < \frac{1}{12}$. Se desea que toda recta \mathcal{L} entre cualesquiera pares de puntos pertenecientes a H_j en la región R_j pase por encima de la región R_i con $i < j$. Consideremos la función $f(x) = m_1(x - j - \Delta) + j^2 - \Delta$, que es la ecuación de una recta que pasa por la esquina inferior derecha de la región R_j , y se desea conocer cuál es el valor máximo de m_1 , tal que la esquina superior izquierda de la región R_i quede por debajo de la recta $f(x)$.

Evaluaremos la función $f(x) = m_1(x - j - \Delta) + j^2 - \Delta$ en la esquina superior izquierda de la región R_i :

$$f(i - \Delta) = m_1(i - j - 2\Delta) + j^2 - \Delta$$

Como se desea encontrar la pendiente máxima que puede tener una recta en una región R_i sin que toque a las demás regiones (ver Figura 3.6), transformaremos la ecuación en una desigualdad. Para simplificar la notación se usara d_{ji} como la diferencia entre j e i .

$$d_{ji} = j - i$$

$$i^2 + \Delta < m_1(-d_{ji} - 2\Delta) + j^2 - \Delta$$

$$i^2 - j^2 + 2\Delta < m_1(-d_{ji} - 2\Delta)$$

$$\frac{j^2 - i^2 - 2\Delta}{d_{ji} + 2\Delta} > m_1$$

Analizaremos esta desigualdad en el caso extremo $i = 0$ y $j = 1$, en el que las dos regiones son mas cercanas en y y por lo tanto la tolerancia de la pendiente m_1 es la menor:

$$\frac{j^2 - i^2 - 2\Delta}{d_{ji} + 2\Delta} > \frac{1 - 2\Delta}{1 + 2\Delta} > \frac{1 - \frac{1}{6}}{1 + \frac{1}{6}}$$

$$1 - \sqrt{\frac{1}{6}} > m_1$$

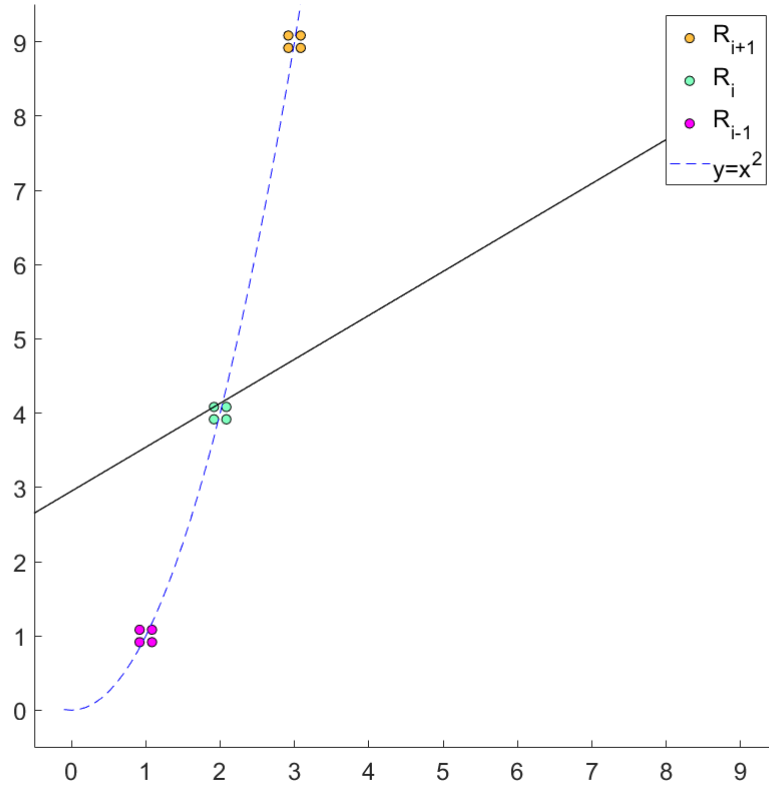


Figura 3.6: Se muestra una recta con pendiente máxima tal que deje por arriba a todas las regiones R_{i+k} y por debajo todas las regiones R_{i-k} .

Finalmente, recordemos que la pendiente máxima dentro de un artefacto H_i es $m_h = \frac{45}{400 * n^5 m^5}$ y como $m_h < 1 - \sqrt{\frac{1}{6}}$ entonces se tiene que la recta \mathcal{L} entre cualquier

par de puntos pertenecientes a H_i pasa por encima de todos los puntos de los artefactos con un índice i con $i < j$; además, en forma parecida se puede probar que la recta \mathcal{L} pasa por debajo de todos los puntos de los artefactos con un índice k con $j < k$. \square

Corolario 3.1.1.1. *Toda recta separadora de un artefacto de tipo horizontal obedece el lema 3.1.1.*

Lema 3.1.2. *Para todo artefacto V_j y toda recta \mathcal{L} entre un par de puntos pertenecientes a V_j se tiene que la recta \mathcal{L} pasa por encima de todos los puntos de los artefactos con un índice k con $j < k$; además, se tiene que la recta \mathcal{L} pasa por debajo de todos los puntos de los artefactos con un índice j con $j < k$.*

Demostración. Supongamos regiones cuadradas R_i que cubren los puntos de cada artefacto y están delimitadas por los puntos $(i-\Delta, i^2+\Delta)$, $(i+\Delta, i^2+\Delta)$, $(i+\Delta, i^2-\Delta)$ y $(i-\Delta, i^2-\Delta)$ donde $\Delta < \frac{1}{12}$. Se desea que toda recta \mathcal{L} entre cualesquiera pares de puntos pertenecientes a V_j en la región R_j pase por encima de la región R_k con $j < k$. Consideremos la función $g(x) = m_2(x-j-\Delta) + j^2 - \Delta$, que es la ecuación de una recta que pasa por la esquina inferior derecha de la región R_j , y se desea conocer cuál es el valor mínimo de m_2 tal que la esquina superior izquierda de la región R_k quede por debajo de la recta $g(x)$:

Evaluaremos la función $g(x) = m_2(x-j-\Delta) + j^2 - \Delta$ en la esquina inferior derecha de la región R_j :

$$g(k-\Delta) = m_2(k-j-2\Delta) + j^2 - \Delta$$

Como se desea encontrar la pendiente mínima que puede tener una recta en una región R_j sin que toque a las demás regiones (ver Figura 3.7), transformaremos la ecuación $g(k-\Delta)$ en una desigualdad. Para simplificar la notación se usará d_{kj} como la diferencia entre k e j .

$$d_{kj} = k - j$$

$$g(k - \Delta) = m_2(k - j - 2\Delta) + j^2 - \Delta$$

$$k^2 + \Delta < m_2(k - j - 2\Delta) + j^2 - \Delta$$

$$k^2 - j^2 + 2\Delta < m_2(d_{kj} - 2\Delta)$$

$$\frac{k^2 - j^2 + 2\Delta}{d_{kj} - 2\Delta} < m_2$$

Analizaremos esta desigualdad en el caso extremo $k = 2n$ y $j = 2n - 1$, en el que las dos regiones son mas cercanas en x y por lo tanto la tolerancia de la pendiente m_2 es la menor:

$$\frac{4n^2 - (4n^2 - 8n + 1) + 2\Delta}{1 - 2\Delta} < m_2$$

$$\frac{8n - 1 + 2\Delta}{1 - 2\Delta} < \frac{8n - 1 + \frac{1}{6}}{\frac{5}{6}} < m_2$$

$$\frac{48n - 6 + 1}{5} < 16n - 1 < m_2$$

$$16n < m_2$$

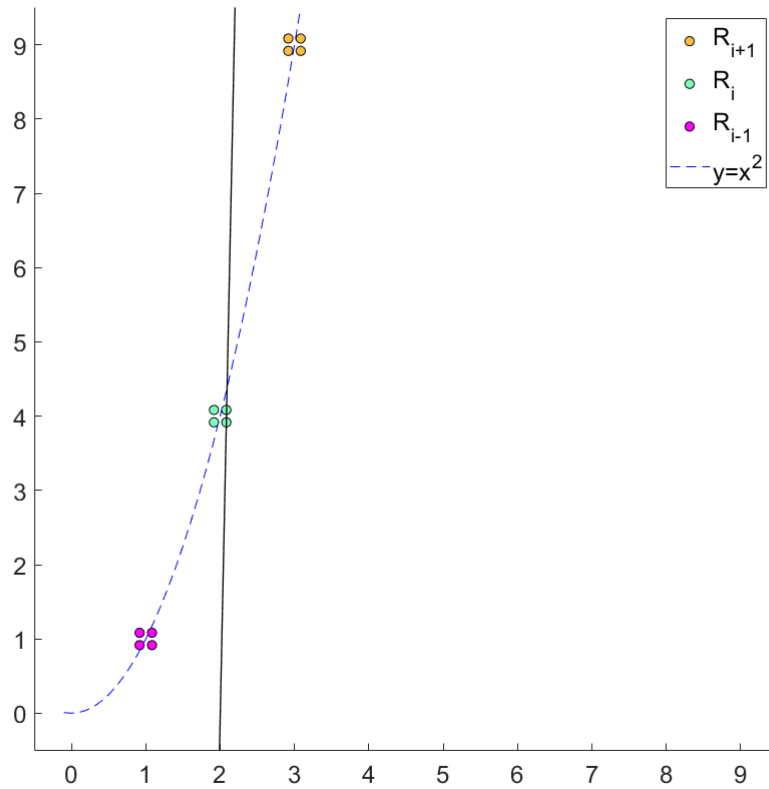


Figura 3.7: Se muestra una recta con pendiente mínima tal que deje por debajo a todas las regiones R_{i+k} y por arriba todas las regiones R_{i-k} .

Finalmente, recordemos que la pendiente mínima dentro de un artefacto V_i es $m_v = \frac{200 \cdot n^5 m^5}{16}$ y como $m_v > 16n$ entonces se tiene que la recta \mathcal{L} entre cualquier par de puntos en V_i pasa por encima de todos los puntos de los artefactos con un índice k con $j < k$; además, en forma parecida se puede probar que la recta \mathcal{L} pasa por debajo de todos los puntos de los artefactos con un índice i con $i < j$. \square

Corolario 3.1.2.1. *Toda recta separadora de un artefacto de tipo vertical obedece el lema 3.1.2.*

Observación 3.1.3. *El conjunto de puntos generados por la reducción Φ está en posición general.*

Demostración. Los lemas 3.1.2 y 3.1.1 tienen como corolario que no existen 3 puntos colineales entre 2 artefactos, por lo que para mostrar que todos los

puntos colocados están en posición general resta probar que no existen 3 puntos colineales entre 3 artefactos. Supongamos regiones cuadradas R_i que cubren los puntos de cada artefacto y están delimitadas por los puntos $(i - \Delta, i^2 + \Delta)$, $(i + \Delta, i^2 + \Delta)$, $(i + \Delta, i^2 - \Delta)$ y $(i - \Delta, i^2 - \Delta)$ donde $\Delta < \frac{1}{12}$. Consideremos una recta formada con puntos en las regiones R_i , R_j y R_k , donde $i < j < k$ para probar que la recta formada pasa por las dos esquinas inferiores derechas de las regiones R_i y R_k deja por debajo a la región R_j (ver Figura 3.8).

Se considerará que los índices i, j, k tienen un orden $i < j < k$, y las diferencias entre estos se denotaran:

$$d_{ji} = j - i$$

$$d_{ki} = k - i$$

$$d_{kj} = k - j$$

Se evaluará la recta $h(x) = \frac{k^2 - i^2}{k - i}(x - k - \Delta) + k^2 - \Delta$ en la esquina superior izquierda de la región R_j , para mostrar que esta queda por debajo. Simplificando:

$$h(x) = (k + i)(x - k - \Delta) + k^2 - \Delta$$

Sustituyendo y evaluando:

$$j^2 + \Delta < (k + i)(j - k - 2\Delta) + k^2 - \Delta$$

$$j^2 + \Delta < (2i + d_{ki})(-d_{kj} - 2\Delta) + k^2 - \Delta$$

$$j^2 + \Delta < (2(k - d_{ki}) + d_{ki})(-d_{kj} - 2\Delta) + k^2 - \Delta$$

$$j^2 + \Delta < (2k - d_{ki})(-d_{kj} - 2\Delta) + k^2 - \Delta$$

$$j^2 + \Delta < k^2 - 2(d_{kj} + 2\Delta)k + d_{ki}(d_{kj} + 2\Delta) - \Delta$$

$$(k - d_{kj})^2 + \Delta < k^2 - 2(d_{kj} + 2\Delta)k + d_{ki}(d_{kj} + 2\Delta) - \Delta$$

$$k^2 - 2d_{kj}k + d_{kj}^2 + \Delta < k^2 - 2(d_{kj} + 2\Delta)k + d_{ki}(d_{kj} + 2\Delta) - \Delta$$

$$d_{kj}^2 + \Delta < d_{ki}(d_{kj} + 2\Delta) - 4k\Delta - \Delta$$

$$d_{kj}^2 + \Delta < d_{ki}(d_{kj} + 2\Delta) - 4k\Delta - \Delta$$

$$d_{kj}^2 + \Delta < d_{ki}(d_{kj} + 2\Delta) < d_{ki}(d_{kj} + 2\Delta) - 4k\Delta - \Delta$$

$$\begin{aligned}
d_{kj}^2 + \Delta &< d_{ki}(d_{kj} + 2\Delta) \\
d_{kj}^2 + \Delta &< (d_{kj} + d_{ji})(d_{kj} + 2\Delta) \\
d_{kj}^2 + \Delta &< d_{kj}^2 + 2d_{kj}\Delta + d_{ji}(d_{kj} + 2\Delta) \\
\Delta &< 2d_{kj}\Delta + d_{ji}(d_{kj} + 2\Delta)
\end{aligned}$$

Nótese que el anterior caso cubre los 2 escenarios simétricos restantes. Si tenemos una recta entre la esquina inferior derecha R_i y la esquina superior izquierda R_j y buscamos probar que se deja por arriba a la esquina inferior derecha de la región R_k , definitivamente su pendiente es menor que la formada en el caso anterior, por lo que se puede concluir que siempre se va a cumplir este segundo escenario. Además, si tomamos a la recta formada por la esquina inferior derecha de la región R_k y la esquina superior izquierda de la región R_j y buscamos probar que se deja por arriba la región R_i por simetría es equivalente al escenario 2. Entonces se tiene que los puntos colocados por la reducción Φ están en posición general. \square

3.1.3. Estructura del polígono arcoíris inducido por la reducción Φ

En esta sección se mostrará que de existir un polígono arcoíris de $2n+2$ aristas sobre los puntos creados por la reducción Φ , este tiene la siguiente estructura: por cada uno de los $2n+2$ artefactos de la reducción, el polígono tiene una y solo una arista que separa dicho artefacto, además estas aristas alternan entre segmentos casi horizontales y segmentos casi verticales. También se probará que el polígono es simple, para lo cual se mostrará que, los vértices del polígono están contenidos en regiones definidas, inducidas por las ubicaciones de los artefactos.

Lema 3.1.3. *La familia de rectas Q_k que separa los puntos de un artefacto arbitrario G_k , ya sea de tipo horizontal, vertical o de cierre, es disjunta con cualquiera otra familia de rectas Q_l separadoras de los puntos del artefacto G_l .*

Demostración. Existen 2 tipos de artefactos, los verticales y los horizontales, los artefactos horizontales fueron escalados en su eje y por un factor muy pequeño, de tal manera que la pendiente máxima de las rectas de Q_k es $m = \frac{37}{1300 * n^5 m^5}$ y por

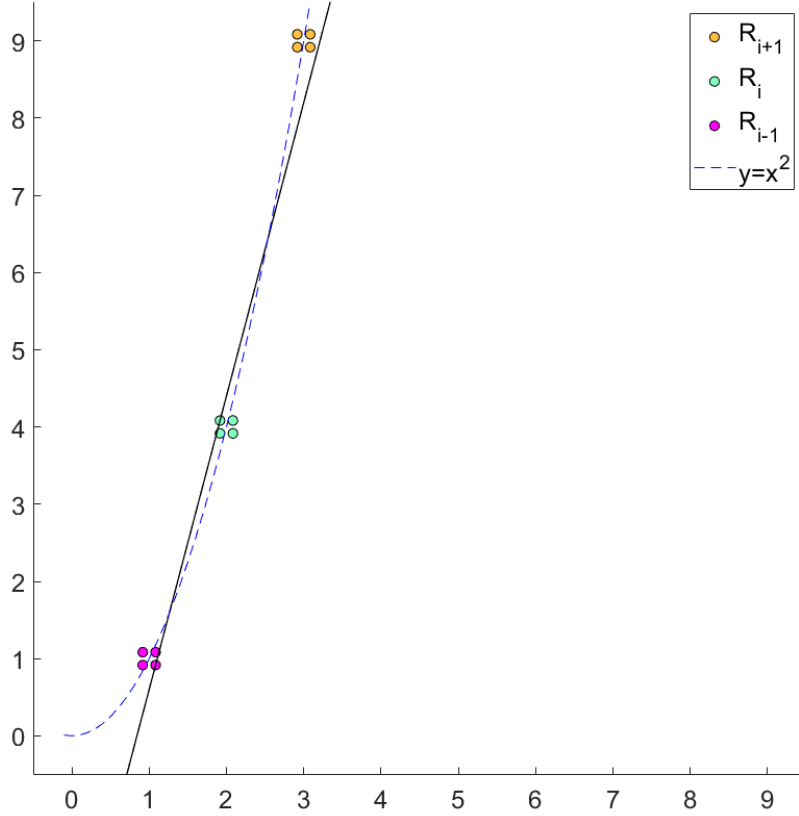


Figura 3.8: Recta con pendiente máxima para evitar que sean colineales 3 puntos entre 3 regiones.

las cotas obtenidas en los lemas 3.1.2 y 3.1.1 sabemos que ninguna recta en Q_k interseca a los cierres convexos de otros artefactos. Para los artefactos de cierre vertical se tiene que la recta con la pendiente mínima perteneciente a Q_k es de $m = 25 * n^5 m^5$; por lo que por las cotas obtenidas en la observación 3.1.3 sabemos que ninguna recta en Q_k interseca a los cierres convexos de otros artefactos. Por lo tanto, ninguna recta de Q_k de tipo horizontal o vertical puede separar los puntos de otro artefacto Q_l . Se sigue que $Q_k \cap Q_l = \emptyset$. \square

Lema 3.1.4. *El polígono formado por intersecciones consecutivas entre las rectas separadoras de los artefactos H_i , V_i , E_h y E_v forman un polígono simple.*

Demostración. Supongamos regiones cuadradas R_i que cubren los puntos de cada artefacto y están delimitadas por los puntos $(i - \Delta, i^2 + \Delta)$, $(i + \Delta, i^2 + \Delta)$, $(i + \Delta, i^2 - \Delta)$ y $(i - \Delta, i^2 - \Delta)$ donde $\Delta < \frac{1}{12}$. Se busca encontrar puntos que delimiten la posición

extrema que podría tomar una intersección entre 2 regiones consecutivas, por los lemas 3.1.2 y 3.1.1 sabemos que cada familia de rectas separadoras de cada región tiene un límite superior de pendiente posible para artefactos horizontales m_h y un límite inferior de pendiente posible para artefactos verticales m_v . Para cada región perteneciente a un artefacto de tipo H_j colocaremos una recta con pendiente m_h en la esquina superior izquierda de dicha región y de manera similar colocaremos una recta con pendiente $-m_h$ en la esquina inferior izquierda de la región. Para cada región de artefactos perteneciente a un artefacto de tipo V_j se colocará una recta con pendiente m_v en la esquina superior izquierda de la región y una recta con pendiente $-m_v$ en la esquina superior derecha de la región (ver Figura 3.9). Las intersecciones entre estas cuatro rectas delimitan las regiones en donde puede ocurrir una intersección entre rectas separadoras.

Nótese que para 2 artefactos consecutivos V_j y H_{j+1} la intersección entre rectas separadoras de estos artefactos ocurre por debajo y a la izquierda del artefacto V_{j+1} . Por lo que estas intersecciones dan como resultado una estructura de apariencia escalonada. De tal manera que si se colocan las líneas separadoras de los artefactos E_h y E_v se tiene que ninguna intersección entre los artefactos V_j y H_{j+1} las atraviesa, por lo que al no tener cruces entre sus aristas, el polígono formado por intersecciones entre las rectas separadoras es un polígono simple. □

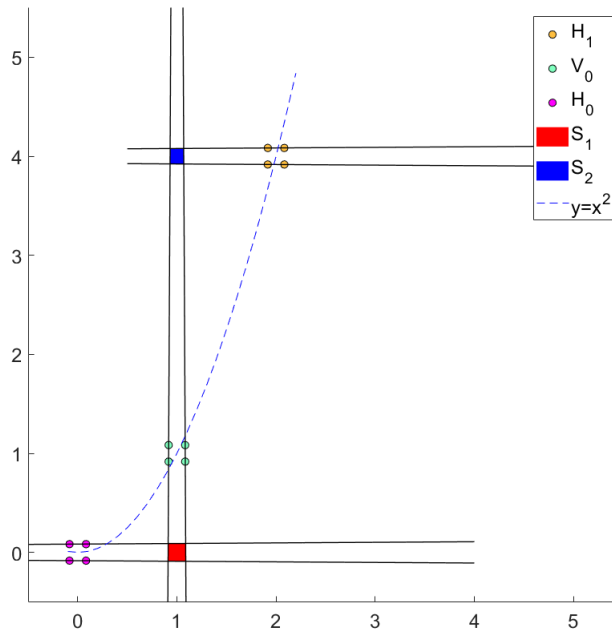


Figura 3.9: Las regiones S_0 y S_1 son aquellas donde es posible tener una intersección entre las rectas separadoras de los artefactos H_0 y V_0 (respectivamente V_0 y H_1).

3.1.4. Prueba de NP-completitud

Lema 3.1.5. *El problema del polígono arcoíris está en NP*

Demostración. Un certificado consiste en los vértices o las aristas que componen al polígono arcoíris. Se puede verificar por cada punto en tiempo $O(q)$ si un punto está dentro de un polígono, lanzando un rayo vertical infinito y verificando el número de aristas que interseca. Por lo tanto en tiempo $O(p \cdot q)$ se puede verificar cada uno de los puntos, determinar cuáles de ellos están dentro del polígono arcoíris y además corroborar que los puntos dentro del polígono sean de colores mutuamente distintos. □

Lema 3.1.6. *La reducción Φ es de tiempo polinomial en el tamaño de una instancia F del problema 1-IN-3-SAT.*

Demostración. Obsérvese que la creación de un artefacto toma tiempo constante

y agregar puntos de cláusula correspondientes a ese artefacto de variable toma tiempo lineal en el número de cláusulas $\Theta(m)$. Además todas las transformaciones, a cada punto de un artefacto, toman tiempo constante, por lo que al hacer ésto por cada variable la reducción tiene una complejidad de $\Theta(n * m)$. \square

Lema 3.1.7. *Sea F una expresión en forma normal conjuntiva, con n variables, m cláusulas y una asignación de variables que satisface a F (en forma 1-IN-3). Entonces es posible generar un polígono arcoíris, de $2n + 2$ aristas, usando el conjunto de puntos de colores producido por la reducción $\Phi(F)$.*

Demostración. Por cada variable X_i si tiene valor *verdadero* entonces dibujamos una recta separadora que cruce por abajo de la región T_i^t en el artefacto H_i y de manera análoga si X_i tiene valor *falso* se dibuja una recta separadora que cruce por abajo de la región T_i^f en el artefacto H_i . Se dibujan las rectas verticales que separen apropiadamente a cada artefacto V_i y se dibujan dos rectas que separen cada artefacto E_h y E_v . Finalmente, las intersecciones de rectas consecutivas (circularmente) serán los vértices del polígono arcoíris y los segmentos entre vértices consecutivos serán sus aristas además sabemos por el lema 3.1.4 que el polígono formado es simple. Por construcción el polígono arcoíris debe contener 8 puntos de color distinto de cada variable, 8 puntos de color distinto de los artefactos E_h y E_v y m puntos de color distinto de las cláusulas. \square

Lema 3.1.8. *Sea F una expresión en forma normal conjuntiva con n variables y m cláusulas. Si existe un polígono arcoíris de $2n+2$ aristas, generado usando un conjunto de $16(n+1) + 3m$ puntos de $8(n+1) + m$ colores distintos producido por la reducción $\Phi(F)$, entonces F es satisfacible (en forma 1-IN-3).*

Demostración. Supongamos que existe un polígono arcoíris generado usando un conjunto de puntos de colores producido por la reducción $\Phi(F)$, entonces cada arista \mathcal{L}_i separa un grupo de puntos de color H_i^a, H_i^b, H_i^c y H_i^d . Por construcción cada arista \mathcal{L}_i deja en el interior del polígono los puntos de la región T_i^t o T_i^f , si la arista deja ahí la región T_i^t , entonces se le asigna el valor *verdadero* a la variable X_i ; de manera análoga si deja la región T_i^f se le asigna el valor *falso* a la variable

X_i . Las regiones delimitadas por los triángulos T_i^t y T_i^f se ven afectadas por la decisión binaria de la arista separadora \mathcal{L}_i antes mencionada, ya que la elección de un tipo de arista implica que una región quede por debajo del segmento y la otra por encima de éste, lo que implica una decisión binaria y, debido a que el polígono solamente contiene un punto de cada color C_j entonces cada cláusula C_j^i de F se satisface solamente por una variable X_i . Se sigue que si existe un polígono arcoíris con la estructura propuesta entonces F es satisfacible (en forma 1-IN-3). □

Corolario 3.1.8.1. *Por los lemas 3.1.5, 3.1.6, 3.1.7 y 3.1.8 se concluye que el problema del polígono arcoíris es NP-completo.*

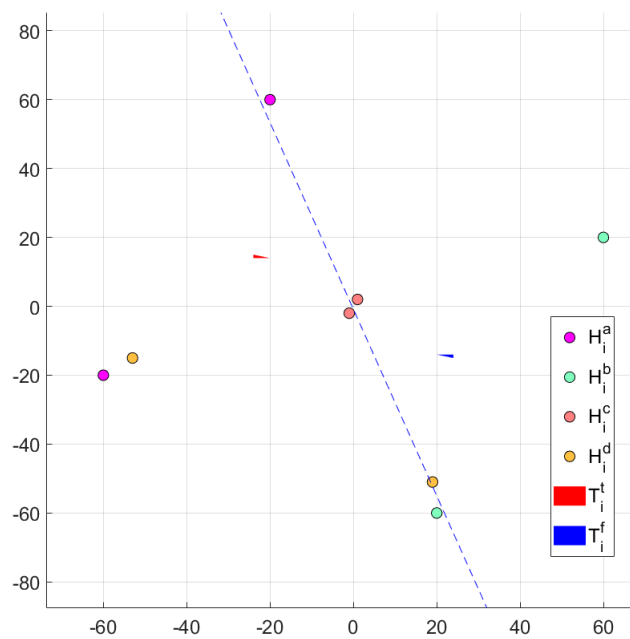


Figura 3.10: Ejemplo de línea separadora en artefacto de variable horizontal.

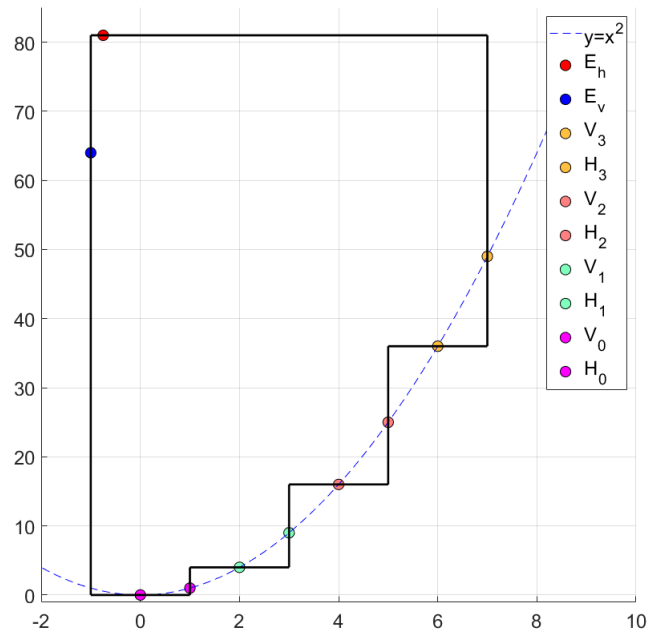


Figura 3.11: Polígono arcoíris resultante de la reducción $\Phi(F)$, con $F = (X_0 \vee \neg X_1 \vee \neg X_2) \wedge (\neg X_1 \vee X_2 \vee X_3) \wedge (X_0 \vee X_1 \vee \neg X_3)$. Cada punto representa un artefacto sin detalle alguno.

CAPÍTULO 4

ÁRBOL ARCOÍRIS

En el artículo [7] se estudia el problema del polígono arcoíris, del cual se deriva el siguiente problema, íntimamente relacionado: dado un conjunto S de p puntos coloreados con r colores y un entero q : ¿es posible crear un árbol arcoíris, con q segmentos de recta, que incluya exactamente r puntos de S de colores distintos? Teniendo en cuenta que un árbol arcoíris es aquel que utiliza exactamente un punto de cada uno de los r colores, de tal manera que los puntos se encuentren en los segmentos de recta. Una de las diferencias entre el problema del árbol arcoíris y el del polígono arcoíris es que este último se da sobre puntos en posición general, mientras que en el problema del árbol arcoíris esto no es un requerimiento.

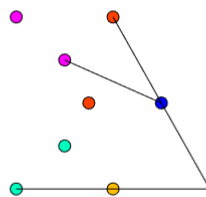


Figura 4.1: Árbol arcoíris de 3 segmentos formado en un conjunto de 8 puntos con 5 colores diferentes.

4.1. El problema del árbol arcoíris es NP completo

4.1.1. Reducción 1-IN-3-SAT al problema del árbol arcoíris

Consideremos la siguiente reducción Φ de instancias 1-IN-3-SAT a instancias del problema del árbol arcoíris. Sea una instancia de 1-IN-3-SAT, donde F es una expresión booleana en forma normal conjuntiva con n variables X_0, X_1, \dots, X_{n-1} y m cláusulas $C_0, C_1, C_2, \dots, C_{m-1}$. La idea general de la reducción Φ es crear un conjunto de puntos $S = S_0 \cup S_1 \cup S_2$ con $r = 2n + m + 3$ colores de tal manera que, de existir, la estructura de todo árbol arcoíris de $n + 1$ segmentos quede forzada a tener un segmento vertical sobre cada una de las rectas verticales $x = 0, \dots, x = n - 1$, y un segmento horizontal sobre la recta $y = 0$, cuidando que estos segmentos cubran exactamente r puntos de colores distintos. Los subconjuntos S_0, S_1 y S_2 se construyen de la siguiente manera: por cada variable X_i se agregaran a S_0 , 4 puntos sobre la recta \mathcal{L}_i , de ecuación $x = i$, en las coordenadas $(i, 1), (i, -1), (i, m + 2), (i, -m - 2)$ con colores X_i, X_i, X'_i, X'_i respectivamente. De tal forma que el árbol arcoíris deberá tener n segmentos verticales, cada uno conteniendo un par de puntos de colores X_i y X'_i , se tomará el par de ordenadas positivas o el otro, dependiendo del valor asignado a X_i . Ahora bien por cada cláusula de la forma $C_j = l_j^a \vee l_j^b \vee l_j^c$, se coloca un punto por cada literal l_j^i en las coordenadas $(i, j + 2)$, donde i es el índice de la variable X_i correspondiente a la literal l_j^i y j es el índice de la clausula C_j , si la variable se encuentra negada entonces en el punto se niega la ordenada quedando $(i, -j - 2)$, por lo que para cada cláusula C_j se agregan 3 puntos de color C_j a S_0 , de tal forma que el árbol arcoíris cubrirá a uno de estos 3 puntos con uno de sus n segmentos verticales. El conjunto S_1 consta de 3 puntos de colores a_0, a_1 y a_2 sobre la recta \mathcal{L}_n de ecuación $y = 0$, en las coordenadas $(n, 0), (n + 1, 0)$ y $(n + 2, 0)$; de tal forma que el árbol tendrá un segmento horizontal que cubra estos puntos y que conecte los segmentos verticales. Finalmente, para evitar la existencia de segmentos indeseados en el árbol y para asegurar que tenga la estructura propuesta, se agrega el conjunto S_2 de puntos *bloqueantes*, por cada par de puntos P_i y P_j en S_0 , que tengan abscisas distintas, se agregarán dos puntos bloqueantes a S_2 de coordenadas únicas de color a_0 ubicados en el segmento $\overline{P_i P_j}$ y evitando que estén sobre las rectas verticales $x = 0, \dots, x = n - 1$ o la recta horizontal $y = 0$. Por cada pareja de puntos P_i en S_0 y P_j en S_1 se agregarán dos puntos bloqueantes de coordenadas únicas de color a_0 ubicados en el segmento $\overline{P_i P_j}$, (ver Figura 4.2).

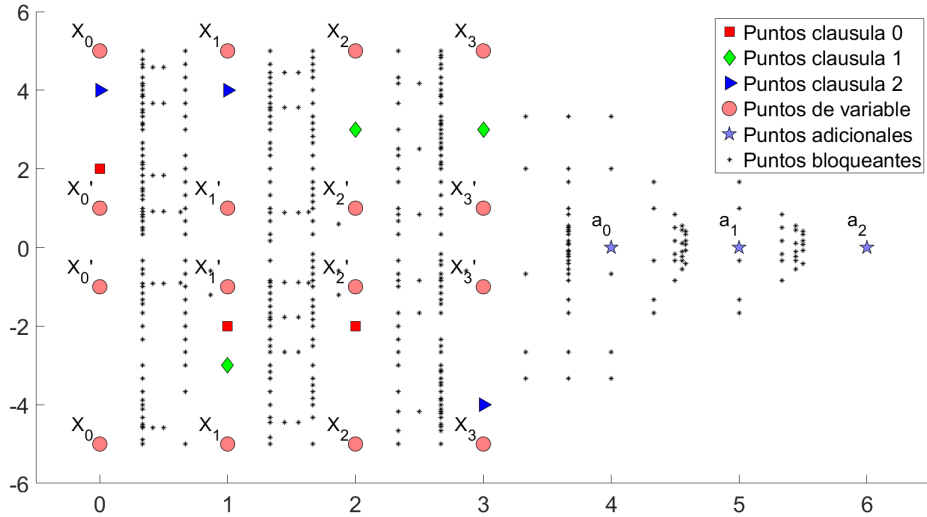


Figura 4.2: Puntos resultantes de la reducción $\Phi(F)$, con $F = (X_0 \vee \neg X_1 \vee \neg X_2) \wedge (\neg X_1 \vee X_2 \vee X_3) \wedge (X_0 \vee X_1 \vee \neg X_3)$. Cabe aclarar que todo punto tiene su color en la etiqueta excepto los puntos bloqueantes que son de color a_0 .

4.1.2. Estructura del árbol arcoíris inducido por la reducción Φ

Lema 4.1.1. *Para toda instancia X de 1-IN-3-SAT satisfacible, todo certificado de $\Phi(X)$ corresponde a un árbol de n segmentos verticales y 1 segmento horizontal, contenidos respectivamente en las rectas verticales $x = 0, \dots, x = n - 1$ y la recta horizontal $y = 0$.*

Demostración. Probemos que de existir el árbol arcoíris de $n + 1$ segmentos, éste debe tener la estructura que propusimos. Para todo $0 \leq i \leq n$, sea Q_i el conjunto de segmentos del árbol que contienen algún punto de $\mathcal{L}_i \cap (S_0 \cup S_1)$. Observemos que todo Q_i es no vacío, si fuera vacío implicaría que hay puntos de algún color sin cubrir, en particular al menos los dos colores que aparecen únicamente en la recta \mathcal{L}_i . Mostremos también que cualesquiera dos conjuntos $Q_i, Q_j, i < j$, son disjuntos: procediendo por contradicción, supongamos que existe un par de conjuntos que no son disjuntos. Sea \mathcal{Z} un segmento en común, \mathcal{Z} cubre un punto p de \mathcal{L}_i , con $i < n$, y por lo tanto de S_0 , y un punto q de \mathcal{L}_j de abscisa distinta a la de p . Por construcción del conjunto de puntos bloqueantes, entre p y q hay al menos dos puntos de color a_0 , con lo cual el árbol no puede ser arcoíris. Se sigue que la suma de las cardinalidades de los Q_i 's es al menos $n + 1$, y observemos que

es a lo más el número de segmentos del árbol, con lo cual es exactamente $n + 1$. Se sigue que para cubrir los (al menos dos) colores de los puntos de $(S_0 \cap S_1)$ en cada \mathcal{L}_i , se requiere que el árbol tenga exactamente un segmento contenido en cada una de las rectas \mathcal{L}_i . \square

4.1.3. Prueba de NP-completitud

Lema 4.1.2. *El problema del árbol arcoíris está en NP*

Demostración. Un certificado consiste en los segmentos de recta que componen al árbol y se puede verificar en tiempo $O(p * q)$ que los segmentos contengan a los puntos del conjunto S y además que solo exista uno de cada color. En tiempo $O(q^2)$ se puede verificar que los segmentos formen un árbol, comparando cada par para saber si todos están conectados y no existe ningún ciclo. \square

Lema 4.1.3. *La reducción Φ es de tiempo polinomial en el tamaño de una instancia F del problema 1-IN-3-SAT.*

Demostración. En la construcción del conjunto de puntos S_0 , por cada variable la construcción toma tiempo lineal en el número de cláusulas $O(m)$, por lo que crear el conjunto S_0 toma tiempo $O(n * m)$. Mientras que la creación del conjunto S_1 es de tiempo constante y la creación del conjunto S_2 toma tiempo cuadrático en el número de puntos ya existentes. Se sigue que la complejidad de la reducción $\Phi(X)$ es $O(n^2 * m^2)$. \square

Lema 4.1.4. *Dada una expresión F en forma normal conjuntiva, con n variables, m cláusulas y una asignación de variables que satisface a F (en forma 1-IN-3), entonces es posible generar un árbol arcoíris, de $n + 1$ segmentos, usando el conjunto de puntos de colores producido por la reducción $\Phi(F)$.*

Demostración. Se construirá un árbol con n segmentos verticales y 1 horizontal. Por cada variable X_i si tiene valor verdadero entonces escogemos el segmento con puntos finales $(i, 0)$ y $(i, m + 2)$, si tiene valor falso entonces escogemos el

segmento con $(i, 0)$ y $(i, -m - 2)$. Observemos que el segmento escogido incluye un punto de color C_j si y solo si la variable X_i en la asignación original hace que la cláusula C_j se satisfaga. Finalmente, agregamos el segmento horizontal con puntos finales $(0, 0)$ y $(n + 2, 0)$, para así cubrir los puntos del subconjunto S_1 generado por Φ . Entonces existe un árbol arcoíris en el conjunto S que tiene la estructura deseada. \square

Lema 4.1.5. *Dada una expresión F en forma normal conjuntiva con n variables y m cláusulas. Si existe un árbol arcoíris de $n + 1$ segmentos, generado usando un conjunto de puntos de colores producido por la reducción $\Phi(F)$, entonces F es satisfacible (en forma 1-IN-3).*

Demostración. Supongamos que existe un árbol arcoíris de $n + 1$ segmentos generado usando un conjunto de puntos de colores producido por la reducción $\Phi(F)$. Por el lema 4.1.1 el árbol tiene n segmentos verticales en las coordenadas $x = 0, 1, \dots, n - 1$ y un segmento horizontal en $y = 0$. Entonces por cada segmento vertical \mathcal{R}_i , si tiene coordenadas positivas en y asignamos el valor verdadero a la variable X_i en F y, si tiene coordenadas negativas en y se le asigna el valor falso a X_i . Nótese que una cláusula libre C_j se satisface por una variable X_i si y solo si el segmento correspondiente \mathcal{R}_i tiene un punto de color C_j . Por lo anterior el árbol solamente contiene un punto de cada color C_j en algún segmento \mathcal{R}_i . Entonces cada cláusula C_j de F se satisface solamente por la variable X_i se sigue que si existe un árbol arcoíris entonces F es satisfacible (en forma 1-IN-3). \square

Corolario 4.1.5.1. *Por los lemas 4.1.2 , 4.1.3, 4.1.4 y 4.1.5 se concluye que el problema del árbol arcoíris es NP-completo.*

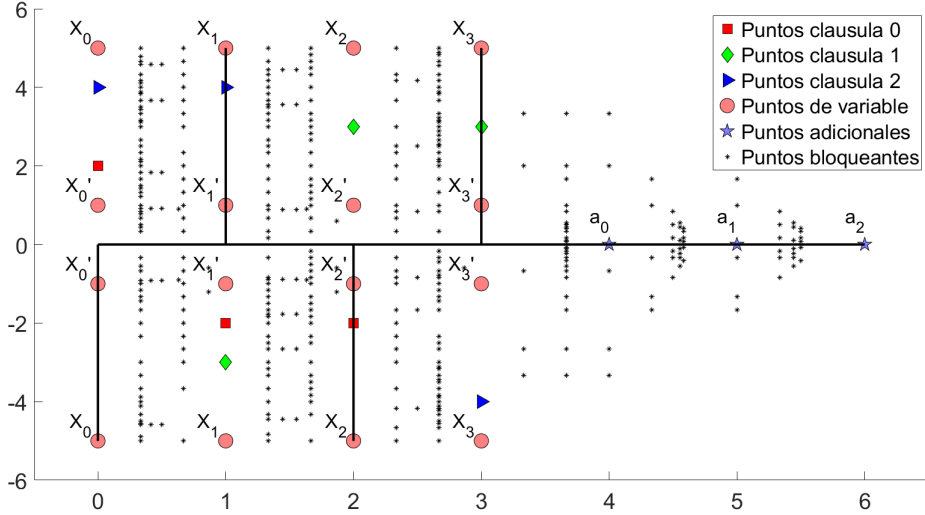


Figura 4.3: Árbol arcoíris generado usando la reducción $\Phi(F)$, con $F = (X_0 \vee \neg X_1 \vee \neg X_2) \wedge (\neg X_1 \vee X_2 \vee X_3) \wedge (X_0 \vee X_1 \vee \neg X_3)$. F se satisface (en forma 1-IN-3) cuando $X_0 = \text{falso}$, $X_1 = \text{verdadero}$, $X_2 = \text{falso}$ y $X_3 = \text{verdadero}$.

4.1.4. Reducción alternativa 1-IN-3-SAT al problema del árbol arcoíris

En la sección 4.1.1 se mostró una reducción de 1-IN-3-SAT al problema del árbol arcoíris en la que se utilizan puntos bloqueantes para forzar la estructura del árbol arcoíris de $n + 1$ segmentos. A continuación se presenta una reducción que no utiliza este tipo de puntos.

Consideremos la siguiente reducción Γ de instancias 1-IN-3-SAT a instancias del problema del árbol arcoíris. Sea una instancia de 1-IN-3-SAT, donde F es una expresión booleana en forma normal conjuntiva con n variables X_0, X_1, \dots, X_{n-1} y m cláusulas $C_0, C_1, C_2, \dots, C_{m-1}$. La idea general de la reducción Γ es crear un conjunto de puntos $S = S_0 \cup S_1$ con $r = (n + 2)(n + 1) + m$ colores de tal manera que, de existir, la estructura de todo árbol arcoíris de $n + 1$ segmentos quede forzada a tener un segmento vertical sobre cada una de las rectas verticales $x = 0, \dots, x = n - 1$, y un segmento horizontal sobre la recta $y = 0$, cuidando que estos segmentos cubran exactamente r puntos de colores distintos. Los subconjuntos S_0, S_1 se construyen de la siguiente manera: por cada variable X_i se agregarán a S_0 , $2n + 4$ puntos sobre la recta \mathcal{L}_i , de ecuación $x = i$, en las coordenadas $(i, 1), (i, m+2), (i, m+3), \dots, (i, m+n+4)$ y $(i, -1), (i, -m-2), (i, -m-3), \dots, (i, -m-n-4)$. Ambas listas con colores $X_i^0, X_i^1, X_i^2, \dots, X_i^{n+1}$. Ahora bien, por cada cláusula de la forma $C_j = l_j^a \vee l_j^b \vee l_j^c$, se colocará un punto por cada variable l_j^i en las coordenadas $(i, j + 2)$ si la variable se encuentra negada entonces en el punto se niega la ordenada, quedando $(i, -j - 2)$, por lo que para cada cláusula C_j se agregan 3 puntos de color C_j a S_0 , de tal forma que el árbol arcoíris deberá cubrir a uno de estos 3 puntos con uno de sus n segmentos verticales, ya sea cubriendo los puntos

con ordenadas positivas o el otro, dependiendo del valor asignado a X_i . S_1 consta de $n + 2$ puntos de colores a_0, a_1, \dots, a_{n+1} , sobre la recta de ecuación $y = 0$, en las coordenadas $(n, 0), (n + 1, 0), \dots, (2n + 1, 0)$, de tal forma que el árbol contenga un segmento horizontal que cubra estos puntos y que conecte los segmentos verticales (ver Figura 4.4).

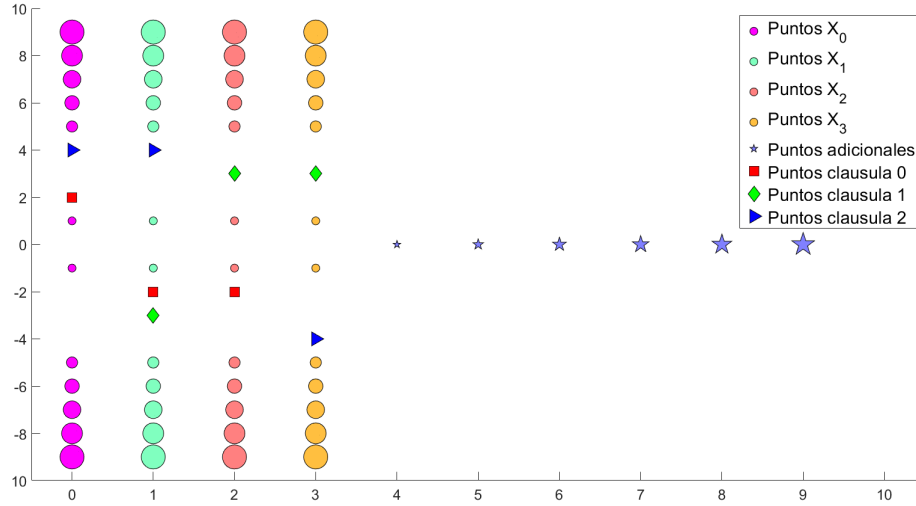


Figura 4.4: Puntos resultantes de la reducción $\Phi(F)$, con $F = (X_0 \vee \neg X_1 \vee \neg X_2) \wedge (\neg X_1 \vee X_2 \vee X_3) \wedge (X_0 \vee X_1 \vee \neg X_3)$.

4.1.5. Estructura del árbol arcoíris inducido por la reducción Γ

Lema 4.1.6. *De existir el árbol arcoíris de $n + 1$ segmentos en el conjunto de puntos creado por la reducción Γ , debe estar formado por n segmentos respectivamente contenidos en las verticales $x = 0, \dots, x = n - 1$ más un segmento contenido en la horizontal $y = 0$.*

Demostración. Observemos que existen dos tipos de segmentos de recta: El tipo 1 está formado por los segmentos contenidos en las verticales $x = 0, \dots, x = n - 1$ y el segmento contenido en la horizontal $y = 0$. El tipo 2 formado por todos los segmentos que no son del tipo 1. Supongamos que se tiene un árbol arcoíris, de $n + 1$ segmentos, T formado por a segmentos del tipo 1 y b segmentos del tipo 2. Cada segmento \mathcal{K} del tipo 2 puede cubrir a lo más $n + 1$ puntos de diferentes colores, ya que a lo más puede intersectar una vez las n rectas $x = 0, \dots, x = n - 1$ y 1 vez la recta $y = 0$. Suponiendo que cada intersección incluye un punto en S y que todos los puntos sobre dicho segmento \mathcal{K} son de colores distintos, entonces los

segmentos de tipo 2 podrán cubrir a lo más $b(n+1)$ puntos de colores distintos. Por construcción cada segmento \mathcal{L} del tipo 1 cubre a lo más $n+2$ puntos de variable y a lo más m puntos de cláusula. Despreciando los puntos de cláusula, se tiene que si T es el árbol arcoíris, de $n+1$ segmentos, por la observación, este debe cubrir al menos $(n+1)(n+2)$ puntos de colores distintos, tomando en cuenta que estos puntos deben cubrirse ya sea por segmentos del tipo 1 o del tipo 2, entonces la igualdad $n^2 + 3n + 2 = a(n+2) + b(n+1)$ debe satisfacerse. Obsérvese que el árbol arcoíris T debe utilizar solo segmentos del tipo 1, de lo contrario si $(b > 0)$ el árbol T no cubriría un punto de cada color o tendría más de $n+1$ segmentos. Entonces todo árbol arcoíris de $n+1$ segmentos sobre el conjunto de puntos de colores S generado por la reducción Γ , tiene la estructura propuesta (ver Figura 4.5). \square

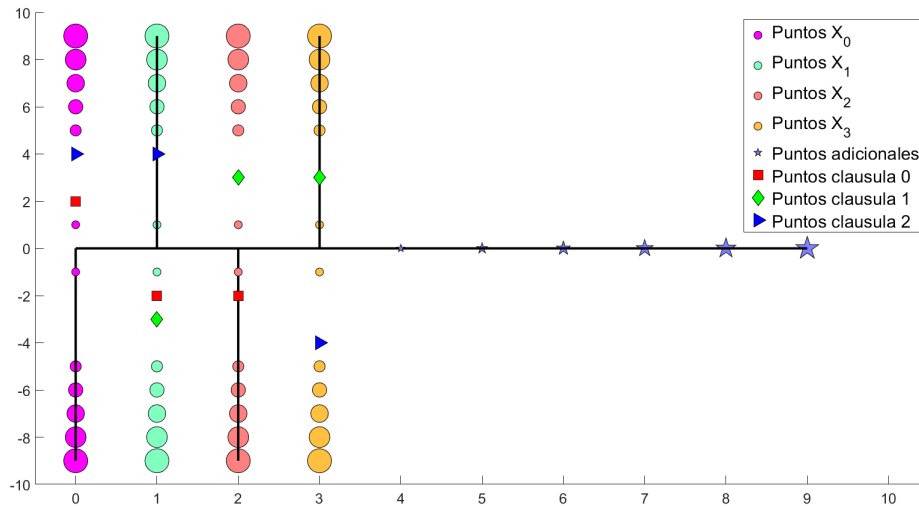


Figura 4.5: Árbol arcoíris generado usando la reducción $\Phi(F)$, con $F = (X_0 \vee \neg X_1 \vee \neg X_2) \wedge (\neg X_1 \vee X_2 \vee X_3) \wedge (X_0 \vee X_1 \vee \neg X_3)$. F se satisface (en forma 1-IN-3) cuando $X_0 = falso$, $X_1 = verdadero$, $X_2 = falso$ y $X_3 = verdadero$.

4.1.6. Prueba de NP-completitud a través de la reducción Γ

Lema 4.1.7. *La reducción Γ es de tiempo polinomial en el tamaño de la instancia F del problema 1-IN-3-SAT.*

Demostración. En la construcción del conjunto de puntos S_0 , por cada variable la construcción toma tiempo lineal en el número de cláusulas $O(m)$, por lo que crear el conjunto S_0 toma tiempo $O(n * m)$. Mientras que la creación del conjunto S_1 es de tiempo $O(n)$. Se sigue que la reducción $\Gamma(X)$ es de complejidad $O(n * m)$. \square

Lema 4.1.8. *Dada una expresión F en forma normal conjuntiva, con n variables, m cláusulas y una asignación a las variables que satisface a F (en forma 1-IN-3), entonces es posible generar un árbol arcoíris, de $n + 1$ segmentos, usando el conjunto de puntos de colores producido por la reducción $\Gamma(F)$.*

Demostración. Se construirá un árbol con n segmentos verticales y 1 horizontal. Por cada variable X_i , si tiene valor verdadero entonces escogemos el segmento con puntos finales $(i, 0)$ e $(i, m + 2)$, si tiene valor falso entonces escogemos el segmento con $(i, 0)$ e $(i, -m - 2)$. Observemos que el segmento escogido incluye un punto de color C_j si y solo si la variable X_i en la asignación original hace que la cláusula C_j se satisfaga. Finalmente agregamos el segmento horizontal con puntos extremos $(0, 0)$ y $(2n + 1, 0)$, para así cubrir los puntos del subconjunto S_1 generado por Γ . Entonces, existe un árbol arcoíris en el conjunto S que tiene la estructura deseada. \square

Lema 4.1.9. *Dada una expresión F en forma normal conjuntiva con n variables y m cláusulas. Si existe un árbol arcoíris de $n + 1$ segmentos, generado usando un conjunto de puntos de colores producido por la reducción $\Gamma(F)$, entonces F es satisfacible (en forma 1-IN-3).*

Demostración. Supongamos que existe un árbol arcoíris generado usando un conjunto de puntos de colores producido por la reducción $\Gamma(F)$, entonces por cada segmento vertical \mathcal{R}_i si tiene ordenadas positivas le asignamos el valor verdadero

a la variable X_i en F y si tiene ordenadas negativas le asignamos el valor falso a X_i . Nótese que una cláusula libre C_j se satisface por una variable X_i si y solo si el segmento correspondiente \mathcal{R}_i tiene un punto de color C_j . Por lo anterior el árbol solamente contiene un punto de cada color C_j en algún segmento \mathcal{R}_i . Entonces cada cláusula C_j de F se satisface solamente por la variable X_i . Se sigue que si existe un árbol arcoíris entonces F es satisfacible (en forma 1-IN-3). \square

Corolario 4.1.9.1. *Por los lemas 4.1.2 , 4.1.7, 4.1.8 y 4.1.9 se concluye que el problema del árbol arcoíris es NP-completo.*

CONCLUSIONES

Se encontró una reducción del problema 3-SAT hacia el problema de los triángulos localmente sin traslape en su variante con puntos de colores (TLST-C), sin embargo, el problema principal permanece abierto puesto que encontrar una reducción desde alguna variante del problema 3-SAT parece ser complicado debido a que solo se cuenta con las posiciones relativas entre los puntos del conjunto de entrada como variable para modelar los artefactos de variable y de cláusula en la reducción.

En contraste en aquellos problemas en los que se encontró una reducción desde alguna variante del problema 3-SAT, además de las posiciones relativas entre los puntos, se tiene el color de los puntos como variable adicional para modelar los artefactos de variable y de cláusula en las reducciones.

Ejemplo de lo anteriormente mencionado es que se encontraron dos reducciones del problema 1-IN-3-SAT hacia los problemas del polígono arcoíris más pequeño y el árbol arcoíris más pequeño mostrando así que ambos son NP-completos, para ambas reducciones se utilizaron las dos variables antes mencionadas para modelar los artefactos de variable y de cláusula.

Lenguaje	Reducción desde:	Dureza
Polígono arcoíris más pequeño.	1-IN-3-SAT	NP-Completo
Árbol arcoíris más pequeño.	1-IN-3-SAT	NP-Completo
Triángulos localmente sin traslape en conjuntos de puntos coloreados.	3-SAT	NP-Completo
Triángulos localmente sin traslape en conjuntos de triángulos.	Conjunto independiente de triángulos	NP-Completo
Triángulos localmente sin traslape.	Desconocido	Desconocida

Cuadro 4.1: Resultados de complejidad del presente trabajo

BIBLIOGRAFÍA

- [1] B. M. Abrego, S. Fernandez-Merchant y D. Flores-Peñaloza. *New bounds on the maximum number of locally non-overlapping triangles in the plane*. 2016.
- [2] O. Aichholzer. <http://www.ist.tugraz.at/aichholzer/research/rp/triangulations/order/types/>. Accessed: 2021-11-21.
- [3] S. Arora y B. Barak. *Computational Complexity: A Modern Approach*. Edición: 1ra. Cambridge University Press, 2009.
- [4] S. A. Cook. “The Complexity of Theorem-Proving Procedures” (1971).
- [5] T. H. Cormen, C. E. Leiserson, R. L. Rivest y C. Stein and. *Introduction to Algorithms*. Edición: 2da. The MIT Press, 2001.
- [6] H. Debrunner y H. Hadwiger. *Combinatorial Geometry in the Plane*. Dover Publications, 1966.
- [7] D. Flores-Peñaloza, M.Kano, L. Martínez-Sandoval, D. Orden, J. Tejel, C. D. Tóth, J. Urrutia y B. Vogtenhuber. “Rainbow polygons for colored point sets in the plane”. *Discrete Mathematics* Volumen 344.Número 7 (2021).
- [8] E. R. Griffor. *Handbook of computability theory*. Edición: 1ra. Elsevier, 1999.
- [9] R. Herken. *The Universal turing machine : a half-century survey*. Kammerer & Unverzagt, 1988.
- [10] A. Pilz y E. Welzl. “Order on Order Types”. *31st International Symposium on Computational Geometry (SoCG 2015)* Volumen 34 (2015), págs. 285-299.

- [11] R. A. Reckhow y S. A. Cook. "Time bounded random access machines". *Journal of Computer and System Sciences* 7.4 (1973), págs. 354-375.
- [12] R. Uehara. "NP-complete problems on a 3-connected cubic planar graph and their application". *Technical Report TWCU-M-0004* (1996).