



UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE CIENCIAS EXACTAS Y NATURALES
DEPARTAMENTO DE COMPUTACIÓN

Sobre grafos amigo-enemigo en el plano con distancia Euclídea

Tesis presentada para optar al título de
Licenciado en Ciencias de la Computación

Matías Exequiel Alvarez

Directora: Dra. Flavia Bonomo

Buenos Aires, 2017

SOBRE GRAFOS AMIGO-ENEMIGO EN EL PLANO CON DISTANCIA EUCLÍDEA

Un grafo con signo es un grafo completo tal que sus aristas están rotuladas con 1, 0 ó -1. Son muy usados actualmente en el contexto de redes sociales, donde los rótulos representan amistad, enemistad o indiferencia.

Una representación de un grafo con signo en un espacio métrico consiste en asignar puntos en el espacio a los vértices de modo que para cada vértice v , la distancia a cualquier vértice w tal que vw es positiva sea estrictamente menor a la distancia a cualquier vértice w tal que vw es negativa. Un grafo se dice amigo-enemigo en un espacio si admite una representación en el mismo. En la recta, el problema de reconocimiento es NP-completo si se admiten aristas nulas, y en el caso en que no hay aristas nulas, el grafo es amigo-enemigo si y sólo si el subgrafo formado por las aristas positivas es un grafo de intervalos unitarios, o, equivalentemente, un grafo de intervalos propio.

En esta tesis se propone estudiar el problema en el plano, con la distancia Euclídea. Se pretende avanzar en una caracterización, encontrando subgrafos inducidos prohibidos y atacando el problema, que parece bastante complejo en el caso general, en subclases de grafos, por ejemplo grafos *split* (cuyo conjunto de vértices puede particionarse en un completo y un conjunto independiente).

Palabras clave: Signed graph, Friend-enemy, Graph drawing, Graph embedding, Line Cluster Embedding.

ABOUT FRIEND-ENEMY GRAPHS ON THE PLANE USING EUCLIDEAN DISTANCE

A signed graph is a complete graph so that its edges are labeled with 1, 0 or -1. They are currently very used in the context of social networks, where labels represent friendship, enmity or indifference.

A representation of a signed graph in a metric space consists of assigning points in space to its vertices so that for each vertex v the distance to any other vertex w , satisfying that vw is positive, results smaller than the distance to any other vertex w which satisfies that vw is negative. A graph is called friend-enemy in a space if it allows a representation on it. On the line, the problem of recognizing whether a graph is friend-enemy is NP-complete if edges labeled zero are allowed, and in the case in which they are not allowed, the graph results in being friend-enemy if and only if the subgraph formed by the positive edges is an unitary interval graph, or equivalently, a proper interval graph.

In this thesis it is proposed to face the problem on the plane with the Euclidean distance. The intention is to go forward on a characterization, finding forbidden induced subgraphs and facing the problem, which appears to be complex in the general case, in graph subclasses for example on *split* graphs (which set of vertices can be partitioned in a complete subgraph and an independent set).

Keywords: Signed graph, Friend-enemy, Graph drawing, Graph embedding, Line Cluster Embedding.

AGRADECIMIENTOS

En primer lugar a toda mi familia; sobre todo a mi hermano Lucas y a mis padres Adrián y Gladys, quienes me bancaron todo este tiempo y desde chico incentivaron mi interés por las ciencias exactas y el gusto enorme por la computación y las matemáticas. A mi abuelo Bepi por el ejemplo de vida del trabajo duro y esfuerzo. A mi abuela Delia, por todas las enseñanzas y los lindos recuerdos que me dejó. A todos mis tíos y a mi abuelo Ernesto. A mi prima Adri.

A Flavia por su enorme paciencia y su interés en dirigir este trabajo. Realmente fue un lujo tenerla como directora. De igual forma agradecer a Willy y a Pia por haber aceptado ser jurados de esta tesis.

A todos mis grandes amigos; en especial a DT y a Lucho por estar siempre aunque los tiempos de estudio complicaban que nos viéramos.

A mi gran amigo Brian, que conocí al comenzar esta carrera y con quien compartí toda su duración. De lo más valioso que me dejó este recorrido.

A Radus por haberme presentado esta carrera en la secundaria y acercarme a la investigación. Sin él, casi con total seguridad, hoy estaría sobre un camino muy distinto.

A Dan por esos ratos divertidos y nerd entre materias. A Jonás, Pato, Leo y todos aquellos con los que me crucé en materias y aportaron algo para que este trabajo fuera posible.

Un agradecimiento especial al DC, lugar que fue mi segunda casa durante estos 5 años y en donde conocí profesores admirables y apasionados por la investigación y las ciencias. De la misma forma a la UBA, por darme la oportunidad de estudiar de forma gratuita y acceder a una educación del más alto nivel, así como a grandes divulgadores que indirectamente me motivaron a interesarme por las ciencias exactas y mantuvieron eso durante toda la carrera, como Adrián Paenza, Diego Golombek, entre otros.

A mi familia. A mis amigos.

Índice general

1..	Introducción	1
1.1.	Contexto y aplicaciones	2
1.2.	Descripción de los temas abordados	3
2..	Grafos <i>friend-enemy</i> sobre \mathbb{R}	4
2.1.	Definiciones y propiedades básicas	4
2.2.	Algunas familias de grafos prohibidos	7
2.3.	Grafos signados completos	11
2.3.1.	Reconocimiento <i>friend-enemy</i> sobre \mathbb{R} , mediante cordalidad	15
2.3.2.	Reconocimiento <i>friend-enemy</i> sobre \mathbb{R} , mediante modelo de intervalos	22
2.4.	Grafos signados genéricos	25
2.4.1.	Análisis de NP-Complejidad	25
2.4.2.	Cota inferior sobre la complejidad temporal	30
2.4.3.	Algoritmo exponencial para el LINE CLUSTER EMBEDDING	33
3..	Grafos <i>friend-enemy</i> sobre \mathbb{R}^2	35
3.1.	Propiedades generales	35
3.2.	Definiciones y propiedades básicas de grafos <i>split</i>	39
3.3.	Caracterización <i>friend-enemy</i> de grafos <i>split</i> por subgrafos prohibidos	40
3.3.1.	Grafos <i>split</i> con clique máximo K_2	40
3.3.2.	Ideas sobre grafos <i>split</i> con clique máximo K_n , con $n \geq 3$	45
4..	Conclusiones y trabajo futuro	50

1. INTRODUCCIÓN

Desde hace muchos años se ha querido representar a las interacciones sociales por medio de relaciones con un modelo y una formalidad que permitiesen inferir propiedades en un grupo grande de personas. Entre estas propiedades, una de las más interesantes es la de definir qué tan amigas son dos personas. Esto, en principio, para abstraer hacia una idea más sencilla, se puede representar como una relación binaria en la cual dichas personas resultan ser o bien amigas, o enemigas; de hecho, esta clase de binarismo existe [1, 2, 16, 17, 18, 19, 26] y la misma idea serviría para otra clase de relaciones habituales como amor-odio, atracción-disgusto, etc.

Kermarrec y Thraves se enfocaron en dicho aspecto y para modelar esto utilizaron grafos signados, sobre los cuales definen la propiedad de drawable (o dibujable), esto es, si a los vértices del grafo les pueden ser asignadas coordenadas de un espacio métrico de forma tal que se cumpla que para todo par de vértices conectados, su distancia resulte menor que la de cada uno de ellos a cualquier otro vértice no conectado. Con esto se obtiene una representación formal de un mapa de relaciones binarias amigo-enemigo, en la que los amigos siempre están más cerca que los enemigos.

Tiempo después, con las propiedades y resultados aportados por dichos investigadores, Cygan, Pipilczuk, Pipilczuk y Wojtaszczyk terminaron por cerrar una gran rama del problema en la cual Kermarrec y Thraves habían incursionado: si es posible representar cualquiera de estos grafos signados (añadiendo la categoría *indiferencia* entre dos vértices) sobre \mathbb{R} y qué complejidad temporal tiene.

El objetivo de este trabajo es el de analizar los avances realizados en esta rama, que fue inaugurada no hace mucho y sobre la cual se tienen resultados sustanciales sólo en la representación sobre la recta, y además aportar algunos resultados sobre la propiedad amigo-enemigo aplicada a la familia de grafos split en el plano Euclídeo.

1.1. Contexto y aplicaciones

Una de las que se cree fue de las primeras aproximaciones a la idea de tener a los amigos más cerca de los enemigos fue la de Cartwright y Harary [3] a mediados de los 50. En este trabajo, su objetivo fue representar formalmente a las relaciones descritas por Heider [10]. Heider planteaba la hipótesis que afirmaba que si una persona \mathcal{P} tiene una cierta relación con otra \mathcal{O} y ésta última tiene una relación con un objeto impersonal \mathcal{X} (siendo ambas relaciones independientes) entonces si llegara a ocurrir que \mathcal{P} no estuviera a gusto con \mathcal{X} , se entraría en una situación de *desbalance* dentro de la unidad relacional $\mathcal{P} - \mathcal{O} - \mathcal{X}$ y emergerían ciertas presiones para que las relaciones cambien de forma de volver a un estado de *balance*. Estas presiones pueden cambiar la relación interpersonal $\mathcal{P} - \mathcal{O}$, la relación de responsabilidad $\mathcal{O} - \mathcal{X}$ o la de evaluación entre $\mathcal{P} - \mathcal{X}$. Las relaciones que describe Heider son la de *gusto* o *disgusto* representado como \mathcal{L} o $\neg\mathcal{L}$ (como $\mathcal{P}\mathcal{L}\mathcal{O}$ o $\mathcal{P}\neg\mathcal{L}\mathcal{O}$) y la de *posesión* o *no pertenencia* representado como \mathcal{U} o $\neg\mathcal{U}$ (como $\mathcal{P}\mathcal{U}\mathcal{X}$ o $\mathcal{P}\neg\mathcal{U}\mathcal{X}$).

En el trabajo de Cartwright y Harary se representaron estas ideas como grafos dirigidos en donde dos vértices pueden tener aristas continuas (relación de *gusto* o valor +1), aristas punteadas (relación de *disgusto* o valor -1) o no tener aristas (relación de *indiferencia* o valor 0). El balance de Heider se formuló sobre relaciones entre tres entidades en donde, por ejemplo, si un ciclo tiene dos relaciones de *gusto* y una de *disgusto*, o todas de *disgusto*, está desbalanceado. Por esto definieron el significado de grafo *balanceado* cuando todo ciclo de cualquier tamaño sea positivo (en donde un ciclo es positivo si y solo si el producto de los valores +1 y -1 de entre todas las aristas que lo conforman es +1). Los grafos *balanceados* fueron utilizados en varios estudios sobre redes sociales [18, 19].

Uno de los ejemplos de uso de grafos signados (o variantes de ellos) se puede ver en [16], en donde se investiga el caso de *Coachsrfing.com*, en el cual se quiere averiguar cuál es la proporción entre la cantidad de personas que aportan su hogar para recibir visitas (*hosts*) y la cantidad de aquellos que se quedan en hogares de otros (*surfers*). Sin ahondar en detalles, esto se puede representar como un grafo dirigido, en donde cada vértice representa una persona con sus características (las cuales los demás miembros pueden ver) y en donde los arcos (v, u) indican un *review* que hizo la persona v sobre u . Estos arcos, pueden tener distintos puntajes (-2 extremadamente negativo, -1 negativo, 0 indiferencia, +1 positivo y +2 extremadamente positivo).

Como otro ejemplo de uso distante de los anteriores, evidencia de aplicaciones a nivel relaciones internacionales se puede apreciar en ejemplos como la Triple Alianza contra la Triple Entente, cuando se aliaron Alemania, el Imperio Austrohúngaro e Italia en contra de la alianza entre Francia, Gran Bretaña y Rusia, y en donde el estado de las relaciones contribuyó contundentemente en el desate de la Primera Guerra Mundial [1, 25].

1.2. Descripción de los temas abordados

En el segundo capítulo se analizará la propiedad *friend-enemy* aplicada a grafos signados sobre \mathbb{R} utilizando la distancia Euclídea. Para esto se iniciará el capítulo aportando las definiciones básicas, algunos ejemplos puntuales de grafos no *friend-enemy* sobre la recta así como otras familias infinitas de grafos no dibujables.

En segundo lugar se verá el estudio de los grafos signados *completos* (aquellos en los cuales todo par de vértices tiene una relación de amistad o de enemistad), sobre los cuales se pueden aplicar herramientas de grafos *cordales* o de grafos de *intervalos* para conseguir un algoritmo lineal que indique si un grafo cumple esta propiedad o no (y además, en caso de cumplirla, que arroje un certificado positivo, es decir, una representación *friend-enemy* sobre \mathbb{R}).

En tercer lugar, se verá que cuando se quiere decidir si la misma propiedad aplica o no sobre grafos signados incompletos o *genéricos* (aquellos en los cuales existen pares de vértices que no tienen aristas y por ende tienen relación de indiferencia), el problema se vuelve NP-Completo. Finalmente, se analiza una cota inferior teórica (asumiendo válida la Hipótesis de Tiempo Exponencial o *Exponential Time Hypothesis*) y finalmente, un algoritmo que resuelve el problema en tiempo $O^*(2^{\epsilon n})$.

En el tercer capítulo se analizará la misma propiedad sobre \mathbb{R}^2 . Se mostrarán algunas propiedades que debe tener todo grafo que sea *friend-enemy* sobre el plano y se pasará a analizar lo que sucede sobre la familia de grafos *split*. El capítulo termina mostrando una caracterización de todos los *split friend-enemy* sobre el plano con clique máxima K_2 y algunas propiedades que se deben cumplir para grafos *split* con clique máxima K_3 en adelante.

Finalmente, en el cuarto capítulo, se presentarán la conclusión e ideas para el trabajo futuro.

2. GRAFOS *FRIEND-ENEMY* SOBRE \mathbb{R}

La idea principal de este capítulo es dar las primeras definiciones sobre el tema y analizar el estado actual de la investigación de la representación de grafos *friend-enemy* sobre \mathbb{R} . Actualmente es un área sobre la cual se pudo avanzar bastante, y se tiene un algoritmo lineal (ver [14]) para la detección de esta propiedad en grafos signados completos. Por último, se analizan otros aportes sobre la detección en grafos signados no completos, la cual resulta un problema NP-completo.

2.1. Definiciones y propiedades básicas

Definición 2.1. Un *grafo signado* es un grafo $G = (V, E)$, tal que el conjunto de aristas se particiona en $E = E^+ \cup E^-$, donde E^+ conforma las aristas positivas (o relaciones de amistad) y E^- las negativas (o de enemistad).

Definición 2.2. Sea $G = (V, E^+ \cup E^-)$ un grafo signado y (M, d) un espacio métrico. Un *friend-enemy drawing* es una función inyectiva $f : V \rightarrow M$, tal que para todos $u, v, w \in V$ tales que $(u, v) \in E^+$ y $(u, w) \in E^-$, se cumple $d(f(u), f(v)) < d(f(u), f(w))$.

Notación 1. Consideramos que un grafo es *friend-enemy* (o que cumple la propiedad *friend-enemy*) cuando admite un *friend-enemy drawing* en el espacio en el cual se trabaja.

En general, a lo largo de esta tesis, el espacio en cuestión será \mathbb{R}^l con $d(\cdot, \cdot)$ siendo la distancia Euclídea. Además, para relajar la notación omitiremos la escritura de f , por lo que hablaremos de la ubicación de v en \mathbb{R}^l obviando a $f(v)$.

Un punto interesante de esto es considerar el conjunto D^l de los grafos que son dibujables en \mathbb{R}^l , para cada $l \in \mathbb{N}$. Luego, es trivial la cadena de inclusiones $D^1 \subset D^2 \subset \dots \subset D^l \subset \dots$. Por lo cual para poder caracterizar de forma ordenada a cada grafo del conjunto de grafos *friend-enemy*, debemos decidir – de existir – cuál es la mínima dimensión l que lo contiene.

A continuación se presenta, como aporte, la demostración formal de la no validez de la propiedad *friend-enemy* sobre los grafos *positive square*, *positive star* y *positive triangle*, que se usan como ejemplo en [14].

Definición 2.3. Llamaremos *negative square* al grafo que tiene cuatro vértices, sus aristas positivas forman un C_4 y sus aristas negativas son las que restan para convertir a C_4 en un K_4 .

Lema 2.4. *El grafo positive square no tiene representación friend-enemy sobre \mathbb{R} .*

Demostración. Nuestro objetivo es mostrar que no existe una asignación de vértices a coordenadas que haga válida la propiedad *friend-enemy*. Etiquetamos a los vértices como se ve en la Fig. 2.1 (a). Dado un vértice cualquiera, sin pérdida de generalidad sea éste a_1 , lo ubicamos sobre la recta. Luego sus vértices adyacentes, a_2 y a_4 , se pueden ubicar de dos formas posibles:

- Uno a cada lado de a_1 . Es decir, el orden quedaría de la forma (a_x, a_1, a_y) con $x, y \in \{2, 4\}, x \neq y$. No puede ocurrir que a_3 esté entre a_x y a_1 , pues de lo contrario valdría $d(a_3, a_1) < d(a_3, a_y)$ donde $(a_3, a_y) \in E^+$ y $(a_3, a_1) \in E^-$ y entonces la representación no sería *friend-enemy*. Análogamente, a_3 tampoco puede estar entre a_1 y a_y , por lo cual debe ser el vértice más a la izquierda o más a la derecha, en dicho orden. Supongamos que queda más a la izquierda, de la forma (a_3, a_x, a_1, a_y) . Entonces valdría $d(a_3, a_1) < d(a_3, a_y)$, con $(a_3, a_y) \in E^+$ y $(a_3, a_1) \in E^-$, lo cual no sería una representación *friend-enemy*. Como esto análogamente aplica con a_3 a derecha (reemplazando y por x), concluimos que este ítem no puede suceder.
- Los vértices a_2 y a_4 quedan ambos a izquierda o ambos a derecha de a_1 . Supongamos que quedan a izquierda (a derecha la prueba es análoga), de la forma (a_x, a_y, a_1) con $x, y \in \{2, 4\}, x \neq y$. Pero $d(a_x, a_y) < d(a_x, a_1)$ con $(a_x, a_1) \in E^+$ y $(a_x, a_y) \in E^-$. Por lo cual, este ítem no puede cumplirse.

Como esas son todas las posibilidades, queda demostrado que el grafo *positive square* no es *friend-enemy* en \mathbb{R} .

□

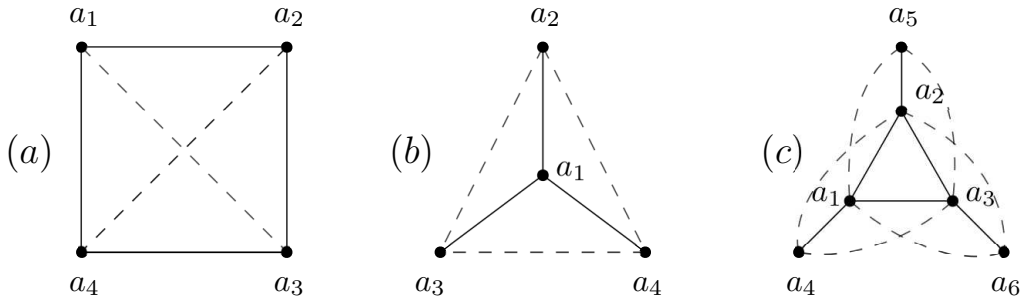


Fig. 2.1: Grafos *positive square* (a), *positive star* (b) y *positive triangle* (c).

Definición 2.5. Llamaremos *positive star* al grafo que tiene cuatro vértices, sus aristas positivas forman un $K_{1,3}$ y sus aristas negativas son las que restan para convertir a $K_{1,3}$ en un K_4 .

Lema 2.6. El grafo *positive star* no tiene representación *friend-enemy* en \mathbb{R} .

Demostración. Partimos la prueba en dos casos posibles de ordenar los vértices sobre la recta, utilizando las etiquetas de los vértices de la Fig. 2.1 (b).

- a_1 aparece más a la izquierda o más a la derecha del resto. Supongamos el primer caso, de la forma (a_1, a_x, a_y, a_z) con $\{x, y, z\}$ una biyección de $\{2, 3, 4\}$. Entonces $d(a_z, a_y) < d(a_z, a_1)$ con $(a_z, a_1) \in E^+$ y $(a_z, a_y) \in E^-$, lo cual no resulta una representación *friend-enemy*. Análogamente, se ve lo mismo con a_1 a derecha.
- a_1 aparece entre medio de (a_x, a_y, a_z) , por lo cual, sin pérdida de generalidad, podemos asumir que el orden es de la forma (a_x, a_1, a_y, a_z) . Entonces $d(a_z, a_y) < d(a_z, a_1)$, con $(a_z, a_1) \in E^+$ y $(a_z, a_y) \in E^-$, lo cual no resulta una representación *friend-enemy*.

Por lo tanto el grafo *positive star* no es *friend-enemy* en \mathbb{R} . □

Definición 2.7. Llamaremos *positive triangle* al grafo que tiene seis vértices y se conforma de la siguiente forma: tres vértices internos formando un K_3 con aristas positivas y tres vértices externos conectándose cada uno a exactamente un vértice distinto del K_3 por medio de una arista positiva y por aristas negativas a los dos vértices restantes del K_3 .

Lema 2.8. *El grafo positive triangle no tiene representación friend-enemy en \mathbb{R} .*

Demostración. En este caso, de haber un orden sobre \mathbb{R} tal que sea una representación *friend-enemy*, debería tener un orden relativo entre los vértices a_1, a_2, a_3 de la forma $(\dots, a_x, \dots, a_y, \dots, a_z, \dots)$, con $\{a_x, a_y, a_z\}$ una biyección de $\{a_1, a_2, a_3\}$.

Tomemos $a_{y'}$ como el único vértice al que se conecta a_y dentro de $\{a_4, a_5, a_6\}$. Luego, no puede ocurrir que en el orden *friend-enemy* $a_{y'}$ quede más a izquierda de a_x , pues resultaría que $d(a_{y'}, a_x) < d(a_{y'}, a_y)$ con $(a_{y'}, a_y) \in E^+$ y $(a_{y'}, a_x) \in E^-$. Análogamente, $a_{y'}$ no puede quedar a derecha de a_z , por lo tanto el orden relativo debe ser de la forma $(\dots, a_x, \dots, a_{y'}, \dots, a_y, \dots, a_z, \dots)$ o $(\dots, a_x, \dots, a_y, \dots, a_{y'}, \dots, a_z, \dots)$.

Si sucediera el primer caso, ocurriría que $d(a_x, a_{y'}) < d(a_x, a_y)$ con $(a_x, a_y) \in E^+$ y $(a_x, a_{y'}) \in E^-$, lo cual no resulta *friend-enemy*. Análogamente, sucedería lo mismo en el segundo caso, con respecto a a_z .

Por lo tanto, el grafo *positive triangle* no resulta *friend-enemy* en \mathbb{R} . □

2.2. Algunas familias de grafos prohibidos

En esta sección mostraremos algunas familias infinitas de grafos presentadas en [15] que no resultan *friend-enemy* sobre \mathbb{R} . Cuando entre dos vértices no se especifique que exista una arista, entonces existirá una relación de indiferencia.

- Familia de grafos \mathcal{F}_1 : sus elementos se notan como $\mathcal{F}_1(n, k)$ y se trata de los grafos de n vértices que conforman un ciclo por medio de aristas positivas y que entre vértices que están a distancia k entre sí (dentro de dicho ciclo) se unen por una arista negativa. Un ejemplo de esto es el *negative square*, representado como $\mathcal{F}_1(4, 2)$.
- Familia de grafos \mathcal{F}_2 : sus elementos se notan como $\mathcal{F}_2(n)$ y se trata de los grafos de $n + 1$ vértices que consisten en un ciclo de n vértices unidos por aristas negativas y un vértice en el centro del ciclo que se une a todos los demás por medio de aristas positivas. Un ejemplo de esto es el *positive star*, representado como $\mathcal{F}_2(3)$.
- Familia de grafos \mathcal{F}_3 : sus elementos se notan como $\mathcal{F}_3(n)$ y se trata de los grafos de $2n$ vértices que, por un lado, consisten en un ciclo de n vértices unidos por aristas positivas. Por otro lado, habrá una biyección entre los vértices del ciclo y los n restantes, en donde cada uno de dichos restantes estará conectado por medio de una arista positiva con el correspondiente del ciclo. Además, estará conectado con aristas negativas con los dos vecinos del vértice del ciclo con el que se conecta de forma positiva. Un ejemplo de esto es el *positive triangle*, representado como $\mathcal{F}_3(3)$.
- Familia de grafos \mathcal{F}_4 : sus elementos se notan como $\mathcal{F}_4(n)$ y es similar a la familia \mathcal{F}_3 pero se invierte el sentido de los valores de las aristas que hay entre los vértices del ciclo y los n externos: cada vértice externo al ciclo de tamaño n (unido, al igual que antes, por aristas positivas) estará conectado por medio de una arista negativa con el vértice que le corresponda en el ciclo. Además, estará conectado con aristas positivas con los dos vecinos del vértice del ciclo con el que se conecta de forma negativa.

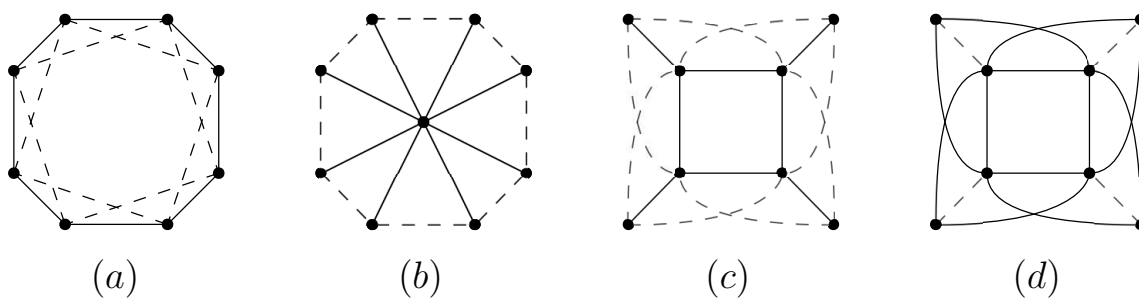


Fig. 2.2: Ejemplos de las familias mencionadas anteriormente: $\mathcal{F}_1(8, 2)$, $\mathcal{F}_2(8)$, $\mathcal{F}_3(4)$ y $\mathcal{F}_4(4)$.

Para probar lo mencionado sobre la primera familia, haremos alusión a la siguiente propiedad demostrada por Pardo et. al. en [22].

Definición 2.9. Dado un grafo signado $G = (V, E^+ \cup E^-)$, se dice que un vértice $x_0 \in V$ es *k-central* si y solo si existen $2k$ vértices y un orden σ sobre \mathbb{R} tal que $\sigma(x_{-k}) < \sigma(x_{-(k-1)}) <$

$\dots < \sigma(x_{-1}) < \sigma(x_0) < \sigma(x_1) < \dots < \sigma(x_{k-1}) < \sigma(x_k)$ y tal que el subgrafo inducido por dichos vértices cumple $(x_i, x_j) \in E^+ \Leftrightarrow |i - j| = 1$ y $(x_i, x_j) \in E^- \Leftrightarrow |i - j| = k + 1$.

Lema 2.10. *Sea $G = (V, E^+ \cup E^-)$ un grafo signado y x_0 un vértice k -central cualquiera (para cierto orden σ). Luego, para todo orden σ' de V sobre \mathbb{R} , que cumpla que para todo $w \in V$ se validen simultáneamente las condiciones:*

- (i) *no existen $v, u \in V$ con $\sigma(v) < \sigma(u) < \sigma(w)$, tales que $(v, w) \in E^+$ y $(u, w) \in E^-$.*
- (ii) *no existen $v, u \in V$ con $\sigma(v) > \sigma(u) > \sigma(w)$, tales que $(v, w) \in E^+$ y $(u, w) \in E^-$.*

existen vértices $u, w \in V$ tales que $\sigma'(u) < \sigma'(x_0) < \sigma'(w)$.

Demostración. Asumamos que σ cumple la hipótesis y por absurdo supongamos que existe un orden σ' que cumple las condiciones (i) y (ii) pero sobre el cual no existen vértices $x_i, x_j \in V$ tales que $\sigma(x_i) < \sigma(x_0) < \sigma(x_j)$. Luego x_0 debe ser el vértice más a la izquierda o más a la derecha en σ' ; sin pérdida de generalidad asumimos que es el de más a la izquierda, pues el caso de estar a la derecha se prueba automáticamente al invertir el orden de las desigualdades de σ' , el cual es otro orden igual de válido para el lema.

Dado que $(x_0, x_1), (x_{-(k-1)}, x_{-k}) \in E^+$ (por ser consecutivos en σ) y $(x_1, x_{-k}) \in E^-$ (por estar a distancia $k + 1$ en σ), debe suceder que $\sigma'(x_1) < \sigma'(x_{-(k-1)})$, pues de lo contrario, si ocurriera $\sigma'(x_1) > \sigma'(x_{-(k-1)})$, cualquier ubicación de x_{-k} se contradiría con (i) y (ii):

- $\sigma'(x_{-(k-1)}) < \sigma'(x_1) < \sigma'(x_{-k})$ implicaría que x_{-k} tiene más cerca a un enemigo (x_1) que a un amigo $\sigma'(x_{-(k-1)})$.
- $\sigma'(x_{-(k-1)}) < \sigma'(x_{-k}) < \sigma'(x_1)$ implicaría que x_1 tiene más cerca a un enemigo (x_{-k}) que a un amigo (x_0 , que se ubica como el más a la izquierda).
- $\sigma'(x_{-k}) < \sigma'(x_{-(k-1)}) < \sigma'(x_1)$ implicaría lo mismo que el caso anterior.

De forma análoga, debe ocurrir $\sigma'(x_{-1}) < \sigma'(x_{k-1})$.

En general debe ocurrir que dado $2 \leq p \leq k - 1$, $\sigma'(x_p) < \sigma'(x_{-(k-p)}) \Rightarrow \sigma'(x_{p+1}) < \sigma'(x_{-(k-(p+1))})$. Como $(x_p, x_{p+1}) \in E^+$ y $(x_{p+1}, x_{-(k-p)}) \in E^-$, debe suceder que $\sigma'(x_{p+1}) < \sigma'(x_{-(k-p)})$. Por otro lado, como $(x_{-(k-(p+1))}, x_{-(k-p)}) \in E^+$ y $(x_{p+1}, x_{-(k-p)}) \in E^-$, debe ocurrir $\sigma'(x_{p+1}) < \sigma'(x_{-(k-(p+1))})$.

Luego, como se cumple $\sigma'(x_p) < \sigma'(x_{-(k-p)}) \Rightarrow \sigma'(x_{p+1}) < \sigma'(x_{-(k-(p+1))})$ para todo $1 \leq p \leq k - 1$, debe ocurrir $\sigma'(x_{k-1}) < \sigma'(x_{-1})$, lo cual no puede suceder pues x_0 está más a la izquierda de todos y es enemigo de x_{k-1} y amigo de x_{-1} . Por lo cual se llega a un absurdo y x_0 no puede quedar en un extremo de σ' . □

Lo que nos dice el lema anterior es que si un vértice v es k -central, no existe una forma de ordenar los vértices de forma que sea una representación *friend-enemy* y al mismo tiempo v sea el menor o el mayor a todos en dicho orden. Sabiendo esto, se demuestra la siguiente propiedad.

Lema 2.11. *Los grafos $\mathcal{F}_1(n, k)$, para todo $n \in \mathbb{N}$ y $2 \leq k \leq \frac{n}{2}$, no tienen representación *friend-enemy* sobre \mathbb{R} .*

Demostración. Dado un grafo $\mathcal{F}_1(n, k)$, se puede ver que todos sus vértices cumplen ser $(k-1)$ -centrales. Por lo tanto, de existir un ordenamiento σ que fuerce una representación *friend-enemy* sobre la recta, tendría que tener un primer y un último elemento (más a izquierda y más a la derecha, respectivamente), pero por el Lema 2.10 esto no puede suceder ya que los elementos de los extremos son $(k-1)$ -centrales. \square

Lema 2.12. *Los grafos $\mathcal{F}_2(2n - 1)$, para todo $n \in \mathbb{N}$, no tienen representación *friend-enemy* sobre \mathbb{R} .*

Demostración. Supongamos que existiera una representación *friend-enemy* para un grafo de $\mathcal{F}_2(2n - 1)$. Luego, si u es el vértice central, debe ocurrir que existan dos vértices consecutivos en el ciclo v_1, v_2 tales que estén ambos a la izquierda o ambos a la derecha de u (si esto no sucediera y todos los vértices estuvieran de forma intercalada uno a cada lado de u , como el ciclo tiene $2n - 1$ vértices ocurriría que el último del ciclo estaría en el mismo lado que el primero). Supongamos que v_1 y v_2 estuvieran a la derecha de u (donde v_1 está entre u y v_2), entonces v_2 tendría más cerca a v_1 (con el que se conecta de forma negativa) que a u (con el que se conecta de forma positiva) y entonces no se trataría de una representación *friend-enemy*. Análogamente sucede si v_1 y v_2 están a la izquierda de u , por lo cual se llega a un absurdo. \square

Para probar las últimas dos propiedades, sobre \mathcal{F}_3 y \mathcal{F}_4 , utilizaremos el siguiente lema auxiliar.

Lema 2.13. *Sea C_n un ciclo impar y sea una representación del mismo sobre \mathbb{R} . Entonces debe existir al menos un vértice que cumpla tener un vecino a su izquierda y otro a su derecha, sobre la recta.*

Demostración. Sea una representación del C_n sobre la recta y supongamos por el absurdo que no existen vértices tales que tengan a cada vecino de cada lado. Por lo cual todo vértice tiene ambos vecinos a su izquierda o ambos a su derecha. Sea un vértice v_1 cualquiera y sea un camino que comience en v_1 y que forme mínimamente un recorrido del ciclo: $(v_1, v_2), (v_2, v_3), \dots, (v_{n-1}, v_n), (v_n, v_1)$. Si marcamos con una L cada vez que una arista (v, u) tiene a v a la izquierda de u y con R cada vez que tiene a v a la derecha de u (lo que, siguiendo el camino, sería *girar a la izquierda* o *girar a la derecha*, respectivamente) entonces ocurre que siempre luego de una L , sigue una R y viceversa (pues siempre que se va desde un lado hacia el otro, se debe volver hacia el lado desde el cual se viene). Por lo tanto, como se marca la misma cantidad de L que de R , el ciclo debe ser par, lo que lleva al absurdo. \square

Lema 2.14. *Los grafos $\mathcal{F}_3(2n - 1)$, para todo $n \in \mathbb{N}$, no tienen representación *friend-enemy* sobre \mathbb{R} .*

Demostración. Supongamos que existiera una representación *friend-enemy* para un grafo de $\mathcal{F}_3(2n - 1)$. Luego, como tiene un ciclo de $2n - 1$ vértices, aplica el Lema 2.10 y debe existir un vértice que tenga un vecino a izquierda y otro a derecha; lo llamamos v . Sean además u_v el vecino fuera del ciclo con el cual se conecta de forma negativa y v_l y v_r los vecinos a izquierda y derecha dentro del ciclo, respectivamente. Supongamos que u_v

estuviera ubicado a izquierda de v en la recta; entonces u_v debe estar más a izquierda que v_l (pues sino no sería una representación *friend-enemy* porque v tendría más cerca a u_v que a v_l). Pero si u_v está más a la izquierda que v_l , luego u_v tendría más cerca a v_l que a v y no sería una representación *friend-enemy*. De forma análoga no puede estar u_v a derecha de v , por lo cual se llega a un absurdo. \square

Lema 2.15. *Los grafos $\mathcal{F}_4(2n - 1)$, para todo $n \in \mathbb{N}$, no tienen representación *friend-enemy* sobre \mathbb{R} .*

Demostración. Supongamos que existiera una representación *friend-enemy* para un grafo de $\mathcal{F}_3(2n - 1)$. Luego, como en la demostración anterior, tiene un ciclo de $2n - 1$ vértices y como aplica el Lema 2.10 debe existir un vértice que tenga un vecino a izquierda y otro a derecha; los llamamos v , v_l y v_r respectivamente. Además, v tiene un vértice fuera del ciclo con el que se conecta por medio de una arista negativa, llamado u_v . Supongamos que u_v estuviera ubicado a la izquierda de v , luego u_v tendría más cerca a v que a v_r y no se trataría de una representación *friend-enemy*. Análogamente, u_v no podría estar a derecha, porque tendría más cerca a v que a v_l ; por lo cual se llega a un absurdo. \square

2.3. Grafos signados completos

Los autores en [14] realizan definiciones y observan las siguientes dos propiedades al querer caracterizar los grafos signados completos *friend-enemy* sobre la recta. Decimos que G es un *grafo signado completo* cuando es de la forma $G = (V, V \times V)$ y $v_{ij} \in \{-1, 1\}, \forall i, j \in \{1, \dots, |V|\}$, con $i \neq j$.

Definición 2.16. Definimos a $v_{l_i^-}$ como el *primer vecino negativo a izquierda* del vértice v_i , esto es, un vecino $v_{l_i^-}$ tal que $v_{l_i^-} < v_i$, $v_{l_i^-} = -1$ y que $\forall v_j$ tal que $v_{l_i^-} < v_j \leq v_i$, ocurre que $v_{ij} \in \{0, 1\}$. Análogamente, se define el *primer vecino negativo a derecha* como $v_{r_i^-}$, tal que $v_i < v_{r_i^-}$, $v_{r_i^-} = -1$ y que $\forall v_j$ tal que $v_i \leq v_j < v_{r_i^-}$, ocurre que $v_{ij} \in \{0, 1\}$.

Definición 2.17. Definimos a $v_{l_i^+}$ como el *último vecino positivo a izquierda* del vértice v_i , esto es, que cumple $v_{l_i^+} < v_i$, $v_{l_i^+} = 1$ y que $\forall v_j$ tal que $v_j < v_{l_i^+} \leq v_i$, ocurre que $v_{ij} \in \{-1, 0\}$. Análogamente, se define el *último vecino positivo a derecha* como $v_{r_i^+}$, que cumple $v_i < v_{r_i^+}$, $v_{r_i^+} = 1$ y que $\forall v_j$ tal que $v_i \leq v_{r_i^+} < v_j$, ocurre que $v_{ij} \in \{-1, 0\}$.

Notación 2. En el caso de la representación sobre \mathbb{R} diremos que dos vértices v, w , los cuales tienen asignadas coordenadas u_v, u_w , cumplen $v < w \iff u_v < u_w$.

Lema 2.18. Sean $G = (V, E^+ \cup E^-)$ un grafo signado y $v_1 < v_2 < \dots < v_n$ un orden del conjunto de vértices V que valida la propiedad *friend-enemy* sobre \mathbb{R} . Entonces para todo v_i se cumple que:

- Dados $v_j < v_i$, $\forall v_{j'} \in V$ tal que $v_{j'} < v_j < v_i$ y $v_{ij} = -1$, se cumple $v_{ij'} \in \{0, -1\}$.
- Dados $v_j > v_i$, $\forall v_{j'} \in V$ tal que $v_{j'} > v_j > v_i$ y $v_{ij} = -1$, se cumple $v_{ij'} \in \{0, -1\}$.

Demostración. Supongamos por el absurdo que σ fuera un ordenamiento de los vértices de G tal que forma una representación *friend-enemy*, pero que no se cumple la condición (a) (la demostración suponiendo que no se cumple (b) es análoga). Luego deben existir vértices $v_{j'} < v_j < v_i$ tales que $v_{ij'} = 1$ pero $v_{ij} = -1$. Pero esto no podría suceder porque es equivalente a decir que $d(v_i, v_{j'}) < d(v_i, v_j)$ con $(v_i, v_{j'}) \in E^+$ y $(v_i, v_j) \in E^-$ y σ no sería una representación *friend-enemy*. □

Lema 2.19. Sea $G = (V, E^+ \cup E^-)$ un grafo signado. Si existe un orden del conjunto de vértices V que cumple simultáneamente las condiciones (a) y (b) del lema anterior, entonces existe un *friend-enemy drawing* de G sobre la recta.

Demostración. Sea G un grafo signado y σ un orden sobre V que cumpla las condiciones (a) y (b), de la forma $\sigma(v_1) < \sigma(v_2) < \dots < \sigma(v_n)$. La idea de la demostración es usar esos valores como coordenadas de forma que se cumpla $u_{v_1} < u_{v_2} < \dots < u_{v_n}$ y finalmente ver que eso es una representación *friend-enemy* sobre \mathbb{R} .

Para conseguir una representación *friend-enemy* válida, basta con ver que se cumplan las siguientes condiciones para todo v_i :

- $d(v_{l_i^+}, v_i) < d(v_{l_i^-}, v_i)$ y $d(v_i, v_{r_i^+}) < d(v_i, v_{r_i^-})$.

$$2. d(v_{l_i^+}, v_i) < d(v_i, v_{r_i^-}) \text{ y } d(v_i, v_{r_i^+}) < d(v_{l_i^-}, v_i).$$

Veamos que se verifica (1). Dado que se cumplen las condiciones (a) y (b), debe ocurrir que para todo v_i valga $\sigma(v_{l_i^-}) < \sigma(v_{l_i^+}) \leq v_i \leq \sigma(v_{r_i^+}) < \sigma(v_{r_i^-})$ y por ende $u_{v_{l_i^-}} < u_{v_{l_i^+}} \leq u_{v_i} \leq u_{v_{r_i^+}} < u_{v_{r_i^-}}$. Por lo cual todo vecino positivo a izquierda se encuentra más cerca de v_i que todo vecino negativo a izquierda, y lo mismo sucede a derecha; por lo tanto se cumple que $d(v_{l_i^+}, v_i) < d(v_{l_i^-}, v_i)$ y $d(v_i, v_{r_i^+}) < d(v_i, v_{r_i^-})$.

Para ver que se cumple (2), basta ver que para todo vértice v_i se cumplan las desigualdades

$$\begin{aligned} u_{v_{r_i^+}} - u_{v_i} &< u_{p_i} - u_{v_{l_i^-}} \\ u_{v_i} - u_{v_{l_i^+}} &< u_{v_{r_i^-}} - u_{p_i} \end{aligned} \quad (2.1)$$

lo que se puede resumir como

$$u_{v_{r_i^+}} - u_{v_{l_i^-}} < 2u_{v_i} < u_{v_{r_i^-}} - u_{v_{l_i^+}} \quad (2.2)$$

Esto se puede garantizar de forma constructiva como lo muestra más adelante el Algoritmo 1. El Teorema 2.20, por otra parte, demuestra que el algoritmo funciona de forma que genera una representación *friend-enemy* de G si la hay, con lo cual se garantiza (2). \square

El algoritmo que aportan Kermarrec y Thraves en [14] usa las condiciones del Lema 2.18 y construye el orden garantizando que para todo vértice v , todo último vecino positivo a izquierda (respectivamente a derecha) quede más cerca de v que todo primer vecino negativo a derecha (respectivamente a izquierda). Esto, que por separado representa las condiciones $u_i - u_{l_i^-} > u_{r_i^+} - u_i$ y $u_{r_i^-} - u_i > u_i - u_{l_i^+}$, les proporciona la desigualdad $u_{r_i^+} + u_{l_i^-} < 2u_i < u_{r_i^-} + u_{l_i^+}$, a la cual abstraen como *condición(i)* en su algoritmo.

La idea es asignar coordenadas u_1, u_2, \dots, u_n a los vértices v_1, v_2, \dots, v_n en orden, de forma que las coordenadas (ya asignadas) de v_1, v_2, \dots, v_{k-1} fueren a la que se le asignará a v_k en la iteración k . Para esto, en dicha iteración, el valor u_k asignado debe satisfacer *condición(j)* para $0 \leq j \leq i-1$. Lo importante de esto es que, como veremos luego del pseudocódigo, de cumplirse las condiciones del Lema 2.18, siempre se podrá en cada paso elegir un u_k , independientemente de los valores escogidos para u_1, u_2, \dots, u_{k-1} .

Cabe destacar que para que esto funcione correctamente se utilizará un vértice ficticio v_0 tal que asegure $v_0 < v_1 < \dots < v_n$, con $v_{i0} = -1$ para todo v_i .

Teorema 2.20. *Sea $G = (V, E^+ \cup E^-)$ un grafo signado completo. Si existe un orden del conjunto de vértices V que cumple simultáneamente las condiciones (a) y (b) del Lema 2.18, entonces el Algoritmo 1 devuelve una asignación de coordenadas a vértices tales que dicha asignación (vista como un orden) cumple las mencionadas condiciones.*

Demostración. Asumiendo válido el antecedente del teorema, sabemos que debemos respetar la desigualdad $u_{r_i^+} + u_{l_i^-} < 2u_i < u_{r_i^-} + u_{l_i^+}$ en todos los u_i para obtener un orden que cumpla (a) y (b). Además, como vimos antes, al momento de asignar el valor de u_k para v_k , este valor queda impuesto por las condiciones de todos los v_j anteriores al v_k actual.

Algorithm 1: Construcción sobre \mathbb{R} de un *friend-enemy drawing* de G

Input : Un grafo $G = (V, E)$, en donde se cumplen las condiciones (a) y (b) del Lema 2.18

Output: Una asignación de coordenadas $\{u_1, u_2, \dots, u_n\}$ a los vértices $\{v_1, v_2, \dots, v_n\}$ de G

```

1  $u_{l_1^-} = 0$ 
2  $\text{condición}(0) = u_1 < u_2 < \dots < u_n$ 
3 for  $i = 1$  to  $n$  do
4   | Setear  $u_i$  tal que satisfaga simultáneamente  $\text{condición}(j)$  para todo  $0 \leq j \leq i - 1$ 
5   |  $i = i + 1$ 
6 end
7 return  $\{u_1, u_2, \dots, u_n\}$ 

```

Queremos ver que en la iteración i , el conjunto de condiciones $\text{condición}(j)$ para $0 \leq j \leq i - 1$ es satisfacible. Supongamos por el absurdo que no, luego debe haber dos condiciones que se contradicen entre sí: sin pérdida de generalidad, para $v_j < v_i$ con $u_{r_i^+} = u_{r_j^-}$ (ya que $u_{r_i^+} < u_{r_j^-}$ es trivial y $u_{r_i^+} > u_{r_j^-}$ es análoga a lo que sigue), por hipótesis $\text{condición}(i)$ hace valer $u_{r_i^+} < 2u_i - u_{l_i^-}$ y $\text{condición}(j)$ hace valer $2u_j - u_{l_j^+} < u_{r_i^-}$, pero suponemos que la asignación produce $2u_i - u_{l_i^-} < 2u_j - u_{l_j^+}$, que es equivalente a $2u_i - 2u_j < u_{l_i^-} - u_{l_j^+}$. Por otro lado, al juntar las condiciones del paso i y j obtenemos $2u_i - 2u_j < u_{l_i^-} + u_{r_i^+} - u_{l_j^+} - u_{r_j^-}$, que (usando que $u_{r_i^+} = u_{r_j^-}$) es equivalente a $2u_i - 2u_j > u_{l_i^-} - u_{l_j^+}$, lo que lleva a un absurdo y entonces $\text{condición}(j)$ para $0 \leq j \leq i - 1$ es satisfacible. \square

Finalmente, lo que acabamos de ver nos dice que el reconocimiento de la propiedad *friend-enemy* sobre la recta se puede realizar sólo determinando si un grafo signado G cumple las propiedades (a) y (b) del Lema 2.18. Lo que vamos a ver a continuación es que cuando esto último sucede, el grafo G^+ resulta cordal, por lo cual se pueden usar algoritmos eficientes en esta clase de grafos.

Definición 2.21. Sea $G = (V, E^+ \cup E^-)$ un grafo signado, entonces su *grafo positivo* correspondiente es el subgrafo inducido por todas las aristas positivas de G . Lo notamos G^+ y observamos que G^+ no resulta un grafo signado.

Definición 2.22. Un grafo G se dice *cordal* si no contiene ciclos inducidos C_k con $k \geq 4$.

Lema 2.23. Sea $G = (V, E^+ \cup E^-)$ un grafo signado completo y G^+ su grafo positivo. Si existe un orden del conjunto de vértices V tal que cada $v \in V$ cumple las condiciones (a) y (b) del Lema 2.18, entonces G^+ es cordal.

Demostración. Supongamos que existe un orden de vértices $v_1 < v_2 < \dots < v_n$ tal que todos cumplen (a) y (b) y que además resulta un orden *friend-enemy* (debe existir tal orden, por Lemas 2.18 y 2.19).

Supongamos por el absurdo que G^+ no fuera cordal, entonces tendría un C_k inducido, con $k \geq 4$, que vistos en el orden anterior resultan $v_{i_1} < v_{i_2} < v_{i_3} < \dots < v_{i_k}$. Como no puede haber una cuerda entre vértices no consecutivos en el ciclo, en particular, debe ocurrir $v_{i_1 i_3} = -1$. Por otro lado, por ser C_k un ciclo, $v_{i_1 i_k} = 1$. Entonces resulta que en nuestro

orden, $d(v_{i_1}, v_{i_3}) < d(v_{i_1}, v_{i_k})$ con $v_{i_1 i_k} \in E^+$ y $v_{i_1 i_3} \in E^-$, lo cual no cumple la propiedad *friend-enemy* y nos lleva al absurdo. \square

Definición 2.24. Una *secuencia (u orden) de eliminación perfecta* es una secuencia de vértices de G (v_1, v_2, \dots, v_n) que cumple que para todo v_i en dicha secuencia, el vecindario $N_{G'}(v_i)$ resulta un subgrafo inducido completo, siendo $G' = G \setminus \{v_1, \dots, v_{i-1}\}$ un subgrafo inducido de G .

Lema 2.25. Sea G un grafo signado completo con más de cuatro vértices y con representación *friend-enemy* en la recta. Si G^+ es conexo, entonces toda secuencia de eliminación perfecta del conjunto de vértices de G satisface, para cada vértice, las condiciones (a) y (b) del Lema 2.18.

Demostración. G tiene representación *friend-enemy* en la recta y por hipótesis G^+ es conexo. Supongamos por el absurdo que hay una secuencia de eliminación perfecta $p_1 < p_2 < \dots < p_n$ de G^+ pero que no se cumple alguna de las condiciones (a) o (b):

- Si no se cumple (a), deben existir vértices tales que $p_i < p_j < p_k$, con $p_{jk} = -1$ y $p_{ik} = 1$, y por ser una secuencia de eliminación perfecta debe también valer $p_{ij} = -1$ (pues de lo contrario en $N(p_i)$ estarían p_j y p_k , los cuales deberían cumplir $p_{jk} = 1$, cosa que contradice nuestra asunción). Como hay al menos cuatro vértices, debe ocurrir que exista un p_x tal que $p_{xj} = 1$ (pues de lo contrario G^+ no sería conexo, pues hasta ahora ni p_i ni p_k se conectan a p_j). Al mismo tiempo, debe ocurrir que $p_{xi} = 1$ ó $p_{xk} = 1$ (pues sino habría dos componentes conexas de 2 vértices cada una). Cualquiera sea el lugar relativo que ocupe p_x entre p_i, p_j y p_k siempre se contradirá que $p_1 < p_2 < \dots < p_n$ es una secuencia de eliminación perfecta, con lo que llegamos al absurdo.
- Si no se cumple (b), la demostración resulta análoga.

Luego todo vértice de toda secuencia de eliminación perfecta de G cumple (a) y (b). \square

Con estos resultados, se puede conseguir un algoritmo que decida si un grafo signado completo G tiene representación *friend-enemy* en función de si el grafo G^+ resulta cordal. Es decir, por un lado, el contrareciproco del Lema 2.23 nos dice que si G^+ no es cordal entonces no existe un ordenamiento de vértices que haga cumplir las condiciones (a) y (b) del Lema 2.18, y por ende, por el Lema 2.19, no existe representación *friend-enemy* en la recta. Por otro lado, si G^+ fuera cordal, se le puede aplicar a cada componente conexa un algoritmo para hallar una secuencia de eliminación perfecta, y por el Lema 2.25 garantizar que cada una es un *friend-enemy drawing* de G sobre \mathbb{R} ; luego se puede juntar todas esas secuencias en una sola, pues al ser cada una de una componente conexa distinta, se mantiene la propiedad.

En la siguiente sección analizaremos la propuesta algorítmica que originalmente presentaron Kermarrec y Thraves en [14], para más adelante pasar a la de Cygan et. al. [4].

2.3.1. Reconocimiento *friend-enemy* sobre \mathbb{R} , mediante cordalidad

La primera versión se basa en un algoritmo llamado LexBFS. Fue propuesto originalmente por Tarjan et. al. en [24] (y luego generalizado por Habib et. al. en [9]) para el reconocimiento de grafos cordales y, en caso de existir, obtener una secuencia de eliminación perfecta. De hecho, hay una equivalencia entre que G sea un grafo cordal y que exista una secuencia de eliminación perfecta.

A continuación veremos algunas propiedades de los grafos cordales que auxilian la demostración de esta última equivalencia mencionada.

Definición 2.26. Dado un grafo $G = (V, E)$, un vértice $v \in V$ se dice *simplicial* si $G[N(v)]$ resulta un grafo completo (donde $N(v)$ refiere a la vecindad de v en G).

Definición 2.27. Sea $G = (V, E)$ un grafo. Dados $v, w \in V$, y $S \subseteq V$, S es un (v, w) -*separador* si v y w están en distintas componentes conexas de $G[V \setminus S]$. En particular, S es un (v, w) -*separador minimal* si para todo $u \in S$, $S \setminus \{u\}$ no es un (v, w) -*separador*.

Lema 2.28. Sea $G = (V, E)$ un grafo cordal. Entonces, para todo (a, b) -*separador minimal* S de G , con $(a, b) \notin E$, sucede que $G[S]$ es completo.

Demostración. En primer lugar, si G es completo, entonces no existe ningún separador para ningún par de vértices. Por lo cual, se cumple trivialmente el lema. Si G no es completo, entonces existen $a, b \in V$ tales que $(a, b) \notin E$, por lo cual existe al menos un (a, b) -*separador minimal* S . Podemos asumir que G es conexo y hay un camino desde a hasta b , porque de lo contrario podríamos tomar a $S = \emptyset$ (el cual es el mínimo entre los tantos minimales que habría, y trivialmente $G[S]$ resultaría un completo). Suponemos que el separador particiona a G en G_1 , S y G_2 . Separamos en dos casos

1. Si $|S| \leq 1$, entonces S está constituido por uno o cero vértices, es decir que $G[S]$ resulta trivialmente completo.
2. Si $|S| \geq 2$, supongamos por absurdo que existen $s_1, s_2 \in S$ tales que $(s_1, s_2) \notin E$. Es fácil ver que s_1 y s_2 tienen cada uno al menos un vecino en G_1 y otro en G_2 , porque de lo contrario S no sería minimal. Entonces tomemos un vecino de cada uno en G_1 y busquemos un camino mínimo que los una dentro de G_1 . Hagamos lo mismo en G_2 . Luego, si unimos esos dos caminos mínimos por medio de s_1 y s_2 , forman un C_k inducido, con $k \geq 4$, lo que nos conduce a un absurdo ya que G es cordal.

□

Lema 2.29. Si $G = (V, E)$ es un grafo completo, tiene al menos un vértice simplicial. Si en cambio G es cordal no completo, entonces tiene al menos dos vértices simpliciales no adyacentes.

Demostración. La primera parte del lema es trivial. Para la segunda parte, aplicamos inducción simple en la cantidad de vértices $|V| \in \mathbb{N}$.

Caso base: Suponemos $|V| = 2$. Como G no es un completo, entonces trivialmente tiene dos vértices simpliciales no adyacentes.

Hipótesis inductiva: Si $|V| = n$ y G es cordal no completo, entonces tiene al menos dos vértices simpliciales no adyacentes.

Paso inductivo: Suponemos $|V| = n + 1$. Partimos la demostración en dos:

1. Si G no es conexo entonces tiene al menos dos componentes conexas G'_1 y G'_2 tales que $|V_{G'_1}|, |V_{G'_2}| < n + 1$, por lo que por hipótesis inductiva cada una tiene al menos un vértice simplicial (pues dichos subgrafos podrían ser completos o no) y como hay al menos dos componentes conexas, G tiene al menos dos vértices simpliciales, no adyacentes.
2. Si G es conexo, al no ser completo, deben existir $a, b \in V$ tales que $(a, b) \notin E$. Entonces existe un (a, b) -separador minimal S que por el Lema 2.28, hace que $G[S]$ sea completo y G quede particionado en G_1 (donde reside a), G_2 (donde reside b) y S . Como $G[V_{G_1} \cup S]$ tiene estrictamente menos de $n + 1$ vértices, por hipótesis inductiva debe tener un vértice simplicial en G_1 , pues o bien $G[V_{G_1} \cup S]$ resulta un grafo completo (y en este caso se puede tomar el simplicial desde G_1), o bien no resulta completo y tiene al menos dos vértices simpliciales no adyacentes (que, como $G[S]$ es completo, uno de ellos debe estar en G_1). La misma idea se aplica en $G[V_{G_2} \cup S]$, por lo cual debe haber al menos otro vértice simplicial en G_2 . Por último, entre estos dos vértices simpliciales no puede haber una arista, ya que de lo contrario G_1 y G_2 estarían conectados por fuera de S .

Por lo tanto, G cordal no completo debe tener al menos dos vértices simpliciales no adyacentes.

□

Lema 2.30. *Si $G = (V, E)$ es un grafo cordal y se le quita un vértice simplicial, se obtiene nuevamente un grafo cordal.*

Demostración. Sea $v \in V$ un vértice simplicial de G (existe por Lema 2.29) y sea $G' = G[V \setminus \{v\}]$. Supongamos por el absurdo que G' no fuera cordal, entonces tendría un C_k inducido, con $k \geq 4$. Ya que G era cordal, no pudo haber tenido tal C_k inducido, por lo tanto lo tuvo que generar el hecho de haber sacado a v . Entonces v estaba conectado con al menos un vértice del C_k , de las siguientes formas posibles:

- Si v estaba conectado a sólo un vértice de C_k , entonces el C_k ya existía en G .
- Si v estaba conectado a dos vértices consecutivos de C_k , resulta igual que el caso anterior, pues v no influía en la existencia de C_k ; por lo tanto éste ya existía en G .
- Si v estaba conectado a dos vértices no consecutivos de C_k , entonces por ser v simplicial ambos vértices compartían una cuerda en C_k (ya que al menos entre v y aquellos dos vértices, deberían formar un K_3), lo cual haría que C_k no sea realmente inducido en G' .
- Si v estaba conectado a tres o más vértices del C_k , entonces por ser v simplicial, por el mismo motivo del punto anterior, tendría que haber una o más cuerdas en C_k .

De todas las formas se llega a un absurdo, por lo cual G' debe ser cordal.

□

Teorema 2.31. *Sea G un grafo. Luego, G es cordal $\Leftrightarrow G$ tiene una secuencia de eliminación perfecta.*

Demostración.

- \Rightarrow) Como G es cordal, tiene al menos un vértice simplicial v_1 (Lema 2.29). Como $G[V \setminus \{v_1\}]$ resulta cordal (Lema 2.30), nuevamente tiene otro vértice simplicial v_2 , el cual se puede quitar y conseguir el grafo cordal $G[V \setminus \{v_1, v_2\}]$. De la misma forma, se puede continuar hasta generar una secuencia (de eliminación perfecta) v_1, v_2, \dots, v_n generada por el orden en que se quitan vértices, tal que para todo v_i en la secuencia vale que v_i es simplicial en $G[V \setminus \{v_1, v_2, \dots, v_{i-1}\}]$.
- \Leftarrow) Supongamos que G no fuera cordal, entonces G tiene un C_k inducido, $k \geq 4$. Sea $\pi = (v_1, v_2, \dots, v_n)$ la secuencia de eliminación perfecta de G de la hipótesis. Sea v_i el primer vértice de C_k en aparecer en π . Como v_i es simplicial en $G[V \setminus \{v_1, \dots, v_{i-1}\}]$ y fue el primero de C_k en aparecer en π , el resto de los vértices de C_k están en $G[V \setminus \{v_1, \dots, v_{i-1}\}]$; en particular sus adyacentes en C_k : v_j y v_k (donde claramente el orden es $i < j, k$). Pero como v_i es simplicial, v_j y v_k deben ser adyacentes, por lo que C_k tendría una cuerda; lo que resulta en un absurdo.

□

En el algoritmo Lexicographic-BFS, el orden queda determinado como en un diccionario. Es decir, queda determinado por el orden a nivel caracteres, tomados de izquierda a derecha en la palabra ($abb < dbb < deb$). De igual forma, si una palabra es prefijo de otra (es decir, una subcadena inicial), entonces la primera, de menor cantidad de caracteres, resulta menor ($abc < abcd$).

Veamos a continuación un pseudocódigo del mismo.

Algorithm 2: Ordenamiento Lexicographic-BFS

Input : Un grafo $G = (V, E)$, con $|V| = n$ y $|E| = m$

Output: Un vector σ de tamaño n conteniendo un orden Lex-BFS

```

1 Vector  $\sigma[n]$ 
2 for  $i \leftarrow 1$  to  $n$  do
3   |  $v_i.label \leftarrow \emptyset$ 
4 end
5 for  $i \leftarrow n$  to 1 do
6   | Elegir un vértice  $v$  de mayor  $label$  sin numerar  $\sigma(i) \leftarrow v$ 
7   | for each  $u \in Ady(v)$ , con  $u$  aún no numerado do
8     | Concatenar  $i$  a la derecha de la etiqueta  $u.label$ 
9   | end
10 end
11 return  $\sigma$ 

```

Para lograr una complejidad temporal eficiente, en [7] se propone utilizar listas de adyacencias sobre G y una cola (internamente como una lista doblemente enlazada) de conjuntos $S_l = \{v \in V \mid label(v) = l, \text{ con } \sigma^{-1}(v) \text{ todavía no definido}\}$ ordenada de forma

lexicográfica de menor a mayor. No se calculan constantemente las etiquetas sino que simplemente se mantiene el orden lexicográfico entre vértices que aún no fueron numerados en σ . Cada conjunto S_l se representa por una lista doblemente enlazada (inicialmente se tendrá sólo a $S_\emptyset = V$ y luego se irá particionando en más conjuntos en función de las etiquetas) y tendrá un atributo *FLAG* que valdrá 0 o 1 (dependiendo de si en la iteración actual se quitó a algún elemento del mismo, y el conjunto está en *revisión*). Usaremos un array $SET(w)$, que dado un w apunta al conjunto que lo contiene: $S_{label(w)}$; y otro array que contenga la dirección de w dentro de $S_{label(w)}$, para que borrar a w de él sea más eficiente. También usaremos un array *FIX LIST*, que contiene punteros a conjuntos, para hacer la limpieza al final de cada iteración y ver si hace falta revisar si algunos conjuntos fueron modificados (aquellos con *FLAG* en 1), pues si en dicho conjunto no quedaron elementos, se los debe quitar de la cola de conjuntos. Por último, también se usarán arrays σ y σ^{-1} .

En las líneas 1-3, se tiene un costo $O(n)$. La línea 6 se puede realizar tomando el primer conjunto de la cola y borrando algún v de $SET(v)$. Luego, la línea 8 se puede realizar creando un nuevo conjunto $S_{l.i}$ por cada conjunto S_l que contenga al menos un $w \in Adj(v)$, borrando todos los w de cada S_l correspondiente (marcando los respectivos *FLAG* en 1) y agregándolos su nuevo $S_{l.i}$ (además de marcando los *FLAG* en 1). Cada $S_{l.i}$ irá ubicado en la cola detrás del $SET(w)$ del cual salió el w en cuestión. Una vez terminado esto con los $w \in Adj(v)$, se realiza el chequeo de limpieza por cada S en *FIX LIST*. Todo esto, que engloba las líneas 6-8, cuesta $O(Adj(v))$, y como se realiza desde n hasta 1, el total cuesta $O(n + m)$.

Finalmente, la complejidad final es $O(n + m)$.

El algoritmo cumple algunas propiedades:

- Si llamamos $L_i(x)$ a la etiqueta que tiene asignado x en la iteración i del algoritmo, entonces dados $j \leq i$, se cumple $L_i(x) \leq L_j(x)$. Esto dice que a medida que se avanzan las iteraciones, las etiquetas pueden agrandarse pero nunca achicarse.
- Para $j < i$, se cumple $L_i(x) < L_i(y) \implies L_j(x) < L_j(y)$. Esto dice que una vez que un vértice quedó con un orden lexicográfico mayor a otro, dicho orden se mantiene hasta el final del algoritmo.
- Si $\sigma^{-1}(w) < \sigma^{-1}(x) < \sigma^{-1}(y)$ e $y \in N(w) \setminus N(x)$, entonces debe existir un vértice $z \in N(x) \setminus N(w)$ con $\sigma^{-1}(y) < \sigma^{-1}(z)$. Esto dice que si $(w, y) \in E$ y en el orden se puso a x luego de y pero antes que w , tiene que ser porque existió anteriormente a y otro vértice z que fue quien le dió prioridad lexicográfica a x por encima de w .

Usaremos este algoritmo para conseguir, en caso de existir, una secuencia de eliminación perfecta. El siguiente teorema prueba lo que buscamos.

Teorema 2.32. *Sea $G = (V, E)$ un grafo. Luego, G es cordal \Leftrightarrow utilizar el algoritmo Lex-BFS sobre G devuelve una secuencia de eliminación perfecta.*

Demostración. La implicación (\Leftarrow) queda probada por el Teorema 2.20.

\Rightarrow) Sea G un grafo cordal y $\{1, 2, \dots, n\}$ los números que usará *Lex-BFS* luego de aplicarse sobre G . Probaremos que $\sigma(1), \sigma(2), \dots, \sigma(n)$ es una secuencia de eliminación perfecta mediante inducción sobre $|V| = n$.

Caso base: Si $|V| = 1$, $\sigma(1)$ es una secuencia de eliminación perfecta.

Hipótesis inductiva: Si G es cordal con $k \leq n$ vértices, entonces el algoritmo aplicado sobre G devuelve una secuencia de eliminación perfecta.

Paso inductivo: Sea $|V| = n + 1$. Como el algoritmo numera en orden inverso (desde n a 1) entonces el último vértice en ser numerado, $\sigma(1)$, debería ser simplicial. Si logramos probar esto, como $\sigma(2), \dots, \sigma(n)$ habría sido la secuencia que arrojaría el algoritmo de no haber existido $\sigma(1)$ en G , luego por hipótesis inductiva valdría que $\sigma(1), \sigma(2), \dots, \sigma(n)$ es una secuencia de eliminación perfecta.

Supongamos que $\sigma(1) = x_0$ no fuera simplicial. Entonces existirían $x_1, x_2 \in N(x_0)$ tales que $(x_1, x_2) \notin E$. Dado que podría haber varios pares de vértices que cumplan eso, nos quedaremos con el par $\{x_1, x_2\}$ que tenga la mayor numeración $\sigma^{-1}(x_2)$. Entonces $\sigma^{-1}(x_0) < \sigma^{-1}(x_1) < \sigma^{-1}(x_2)$, pues x_0 terminó siendo el de menor numeración y por como elegimos a x_2 , el de mayor numeración. También vale que $x_2 \in N(x_0) \setminus N(x_1)$, entonces existe $x_3 \in N(x_1) \setminus N(x_0)$ tal que $\sigma^{-1}(x_2) < \sigma^{-1}(x_3)$; en particular, nos quedaremos con el x_3 de mayor numeración que cumpla esto. Notemos que $(x_2, x_3) \notin E$, pues de lo contrario existiría el C_4 inducido (x_0, x_1, x_3, x_2) y se llegaría a un absurdo pues G no sería cordal. Se puede seguir la idea con un x_4 : como $\sigma^{-1}(x_1) < \sigma^{-1}(x_2) < \sigma^{-1}(x_3)$ y $x_3 \in N(x_1) \setminus N(x_2)$, entonces debe existir un $x_4 \in N(x_2) \setminus N(x_1)$ tal que $\sigma^{-1}(x_3) < \sigma^{-1}(x_4)$. En particular $(x_0, x_4) \notin E$, pues al elegir x_2 habíamos exigido que x_2 fuera el vecino de x_0 de mayor numeración.

En general, sucede que para todo $i \geq 0$, $(x_i, x_{i+2}) \in E$, $\sigma^{-1}(x_i) < \sigma^{-1}(x_{i+1})$, $(x_i, x_{i+3}) \notin E$ y $(x_i, x_{i+4}) \notin E$ (esto último, porque x_{i+2} fue elegido en orden como el vecino de mayor numeración de x_i ; por lo cual x_{i+4} no puede ser vecino de x_i).

Entonces, en este esquema uno está obligado (sobre todo por la última de las propiedades del algoritmo) a necesitar constantemente que exista un nuevo vértice: como $\sigma^{-1}(x_i) < \sigma^{-1}(x_{i+1}) < \sigma^{-1}(x_{i+2})$ y $x_{i+2} \in N(x_i) \setminus N(x_{i+1})$, entonces debe existir $x_{i+3} \in N(x_{i+1}) \setminus N(x_i)$ tal que $\sigma^{-1}(x_{i+2}) < \sigma^{-1}(x_{i+3})$, cumpliendo lo anteriormente comentado. Dado que G era finito, se llega a un absurdo y luego x_0 debe ser simplicial, lo que prueba por lo dicho al principio que $\sigma(1), \sigma(2), \dots, \sigma(n)$ es una secuencia de eliminación perfecta.

□

El primer algoritmo nos devuelve una secuencia de eliminación perfecta si y solo si G resulta cordal. Nos falta un segundo algoritmo para testear si una secuencia es de eliminación perfecta.

Algorithm 3: Testeo de secuencia de eliminación perfecta

Input : Las listas de adyacencias de G y un ordenamiento σ
Output: *True* si y solo si la secuencia es de eliminación perfecta

```

1 for  $v \in V$  do
2   |  $A(v) \leftarrow \emptyset$ 
3 end
4 for  $i \leftarrow 1$  to  $n - 1$  do
5   |  $v \leftarrow \sigma(i)$ 
6   |  $C \leftarrow \{x \in Adj(v) \mid \sigma^{-1}(v) < \sigma^{-1}(x)\}$ 
7   | if  $C \neq \emptyset$  then
8     |  $u \leftarrow \sigma(\min\{\sigma^{-1}(x) \mid x \in C\})$ 
9     |  $A(u) \leftarrow A(u) \cdot (C \setminus \{u\})$ 
10  | end
11  | if  $A(v) \setminus Adj(v) \neq \emptyset$  then
12    | return False
13  | end
14 end
15 return True

```

Teorema 2.33. *El algoritmo de testeo de secuencia de eliminación perfecta (Algoritmo 3) devuelve False sobre $\sigma \Leftrightarrow \sigma$ no es una secuencia de eliminación perfecta.*

Demostración.

\Rightarrow) Si el algoritmo devolvió *false* durante la iteración número $\sigma^{-1}(u)$, es porque ocurrió $A(u) \setminus Adj(u) \neq \emptyset$, es decir, debe existir un $w \in A(u)$ tal que $w \notin Adj(u)$, el cual tuvo que habersele sido agregado a $A(u)$ por medio de un v (en la iteración $\sigma^{-1}(v)$), momento en el que ocurrió simultáneamente $u, w \in Adj(v)$, con $\sigma^{-1}(v) < \sigma^{-1}(u) < \sigma^{-1}(w)$. Esto significa que, dado el orden secuencial entre v, u y w , sucede que v no es simplicial en $G[V \setminus \{\sigma(1), \dots, \sigma^{-1}(v) - 1\}]$, por lo cual resulta correcto que el algoritmo arrojará *false*.

\Leftarrow) Supongamos por el absurdo que el algoritmo arrojará *true* cuando σ no es una secuencia de eliminación perfecta. Sea v el vértice con mayor $\sigma^{-1}(v)$ dentro de los que cumplen que el conjunto $C_v = \{x \in Adj(v) \mid \sigma^{-1}(v) < \sigma^{-1}(x)\}$ (obtenido en la línea 6 de la iteración $\sigma^{-1}(v)$) no generaba un subgrafo completo de G . Sea u el obtenido en la línea 8 de la misma iteración (la $\sigma^{-1}(v)$ -ésima), el cual luego realiza $A(u) \leftarrow A(u) \cdot (C_v \setminus \{u\})$. Sucede entonces que durante la iteración número $\sigma^{-1}(u)$, como no se devuelve *false*, vale que $A(u) \setminus Adj(v) = \emptyset$. Esto implica que debió ocurrir que todo $x \in (C_v \setminus \{u\})$ fuera adyacente a u y que al mismo tiempo todo par de vértices $x, y \in (C_v \setminus \{u\})$ sean adyacentes entre sí; afirmaciones que nos llevan a que C_v tuvo que haber sido un subgrafo completo y finalmente, a un absurdo. □

En este caso se utilizan los arrays σ , σ^{-1} y las listas $Adj(v)$ y $A(v)$. Las líneas 1-3 se ejecutan n veces. Las líneas 5-10, en caso de ejecutarse todas, se pueden calcular en un mismo recorrido de $Adj(v)$, lo cual tiene costo $O(|Adj(v)|)$. En la línea 7, se dará $C = \emptyset$

exactamente $q - 1$ veces, siendo q la cantidad de componentes conexas de G . Finalmente la línea 11 se puede realizar en $O(|Adj(v)| + |A(v)|)$, pues se puede utilizar un array de ceros (uno por cada $w \in V$) que inicialmente cargamos de unos sólo en las posiciones que corresponden a cada $Adj(v)$, y luego de eso, para cada $w \in A(v)$, revisamos si el arreglo en la posición w tiene un cero: con que haya sólo un $w \in A(v)$ que tenga un cero, decimos que $A(v) \setminus Adj(v) \neq \emptyset$. De lo contrario, si todos los de $A(v)$ tenían un uno en nuestro vector auxiliar, volvemos a llenar al vector auxiliar de ceros (para posterior reuso) y decimos que $A(v) \setminus Adj(v) = \emptyset$. Sumando todo esto, obtenemos el costo $O(n + \sum_{v \in V} |Adj(v)| + \sum_{u \in V} |A(u)|)$, pero puesto que la suma del medio tiene más elementos que la última (pues cada $Adj(v)$ es parte de como mucho un elemento $A(u)$), se pueden reemplazar ambas sumas por un costo m .

Finalmente, la complejidad del algoritmo es de $O(n + m)$.

Teorema 2.34. *Se puede saber si un grafo signado completo G es friend-enemy sobre \mathbb{R} en tiempo polinomial. Más aún, se puede conseguir con complejidad $O(n^2)$.*

Demostración. Se sigue de aplicar las propiedades de los Lemas 2.23 y 2.25 y los Teoremas 2.20, 2.32 y 2.33. Por lo visto, se puede conseguir y testear una secuencia de eliminación perfecta en $O(n + m^+)$, y luego corroborar que cumpla las condiciones (a) y (b) y conseguir las coordenadas de la representación *friend-enemy* en $O(n^2)$. \square

2.3.2. Reconocimiento *friend-enemy* sobre \mathbb{R} , mediante modelo de intervalos

La segunda versión que se presentará es la propuesta en [4] por Cygan et. al., que aborda el problema de reconocer si un grafo signado completo G resulta *friend-enemy* sobre la recta desde la perspectiva de los grafos de intervalos. Para esto vemos algunas propiedades y teoremas auxiliares, que se aportan en otros papers.

Definición 2.35. Sean $G = (V, E)$ un grafo simple, \mathcal{F} una familia de tamaño $|V|$ de intervalos sobre \mathbb{R} (intervalos continuos y sin compartir ningún extremo entre distintos intervalos) y $f_{\mathcal{F}} : V \rightarrow \mathcal{F}$ una biyección. Diremos que \mathcal{F} es un *modelo de intervalos* para G si para todo $v, w \in V$, con $v \neq w$, ocurre que $(v, w) \in E$ es equivalente a $f_{\mathcal{F}}(v) \cap f_{\mathcal{F}}(w) \neq \emptyset$.

La definición anterior nos dice que un modelo de intervalos es una biyección de vértices de G a intervalos sobre \mathbb{R} de forma tal que dos vértices son adyacentes en G si y solo si los intervalos que los representan tienen intersección no vacía sobre la recta.

Definición 2.36. Un modelo de intervalos \mathcal{F} se dice *propio* si ningún intervalo está totalmente contenido en otro.

Notación 3. A los grafos que tengan al menos un modelo de intervalos posible se los llamará *grafos de intervalos*, mientras que si tienen al menos un modelo de intervalos propio, se los llamará *grafos de intervalos propio*.

Definición 2.37. Un orden σ sobre el conjunto de vértices de un grafo $G = (V, E)$ se dice un *ordenamiento paraguas* (u *umbrella*) si para cada tripla de vértices que verifica $\sigma(v) < \sigma(u) < \sigma(w)$, se cumple que $(v, w) \in E \implies (v, u), (u, w) \in E$.

Definición 2.38. Un grafo $G = (V, E)$ se dice *grafo de indiferencia* si existe una función $f : V \rightarrow \mathbb{R}$ tal que para todo $v, u \in V$, cumple $(v, u) \in E \Leftrightarrow |f(u) - f(v)| \leq 1$.

En [23], Roberts demuestra la equivalencia entre ciertas clases de grafos de intervalos y grafos de indiferencia.

Teorema 2.39. *Sea G un grafo simple. Luego, son equivalentes:*

- (i). G es un grafo de intervalos propio.
- (ii). G es un grafo de intervalos sin subgrafos *claw* ($K_{1,3}$) inducidos.
- (iii). G es un grafo de indiferencia.

En [21], Olariu demuestra la siguiente equivalencia entre que G sea un grafo de intervalos y la necesidad de la existencia de una versión débil del ordenamiento paraguas.

Teorema 2.40. *Sea un grafo $G = (V, E)$. Luego, G es un grafo de intervalos \Leftrightarrow existe un ordenamiento de V , $v_1 < v_2 < \dots < v_n$ tal que para toda tripla de vértices que cumplan $v_i < v_j < v_k$, validen $(v_i, v_k) \in E \implies (v_i, v_j) \in E$.*

Utilizando este teorema, Looges y Olariu en [20] demostraron lo siguiente.

Teorema 2.41. *Sea $G = (V, E)$ un grafo simple. G es un grafo de indiferencia \Leftrightarrow existe un ordenamiento paraguas de G .*

Demostración.

\Leftarrow) En primer lugar, existe un ordenamiento σ en el que para cada $\sigma(v) < \sigma(u) < \sigma(w)$, se cumple que $(v, w) \in E \implies (v, u) \in E, (u, w) \in E$. Por lo que, por el Teorema 2.30, G es un grafo de intervalos. Necesitamos ver que es un grafo de indiferencias, por lo que probaremos que G es *claw-free* y con el Teorema 2.29 terminará la demostración.

Supongamos por el absurdo que G tuviera un subgrafo *claw* inducido: existirían vértices x, v, u, w tales que $(x, v) \in E, (x, u) \in E$ y $(x, w) \in E$. No puede ocurrir que $\sigma(x) < \min\{\sigma(v), \sigma(u), \sigma(w)\}$ porque por hipótesis deberían entonces existir las aristas $(v, u) \in E, (v, w) \in E, (u, w) \in E$ y $\{x, v, u, w\}$ no formarían un *claw* inducido. Sucedería lo mismo si $\sigma(x) > \max\{\sigma(v), \sigma(u), \sigma(w)\}$, por lo cual x debe estar ubicado entre medio: supongamos sin pérdida de generalidad (pues tanto v como w son indistinguibles) que $\sigma(v) = \min\{\sigma(x), \sigma(v), \sigma(u), \sigma(w)\}$ y $\sigma(w) = \max\{\sigma(x), \sigma(v), \sigma(u), \sigma(w)\}$. Debe ocurrir que x y u cumplan $\sigma(v) < \sigma(u) < \sigma(x) < \sigma(w)$ (en cuyo caso debería existir $(v, u) \in E$) o $\sigma(v) < \sigma(x) < \sigma(u) < \sigma(w)$ (en cuyo caso debería existir $(u, w) \in E$), por lo que siempre se llega a un absurdo y luego G debe ser *claw-free*.

\Rightarrow) Sea G un grafo de indiferencia. Como G es además un grafo de intervalos, definimos el intervalo de cada $v \in V$, como $I_v = [a_v, b_v]$ y un orden $\sigma(v) < \sigma(u) \Leftrightarrow^1 (a_v < a_u)$ ó $((a_v = a_u) \text{ y } (b_v \leq b_u))$. Dados v, u, w tales que $\sigma(v) < \sigma(u) < \sigma(w)$ y $(v, w) \in E$, entonces:

- $(v, u) \in E$: sabemos que $\sigma(v) \leq \sigma(u) \leq \sigma(w)$ (deriva de aplicar ¹) y que $a_w \leq b_v$ (pues $I_v \cap I_w \neq \emptyset$), luego vale $a_u \leq b_v$. Entonces $I_v \cap I_u \neq \emptyset$ y por lo tanto $(v, u) \in E$.
- $(u, w) \in E$: no puede suceder que $I_u \subset I_v$, por lo cual $a_v \leq a_u$ fuerza que valga $b_v \leq b_u$. Luego, $a_w \leq b_u$ y por lo tanto $(u, w) \in E$.

□

Corolario 2.42. G es un grafo de intervalos propio $\Leftrightarrow G$ admite un ordenamiento paraguas.

Demostración. Por Teoremas 2.29 y 2.31.

□

Cygan et. al. proponen combinar los Lemas 2.18 y 2.19 de forma de acercarse a la definición de *ordenamiento paraguas*.

Lema 2.43. Sea $G = (V, E^+ \cup E^-)$ un grafo signado. Luego, G tiene una representación *friend-enemy* en $\mathbb{R} \Leftrightarrow$ existe un ordenamiento σ de V tal que para todo $w \in V$ cumple que

- (i). no existen $v, u \in V$ con $\sigma(v) < \sigma(u) < \sigma(w)$, tales que $(v, w) \in E^+$ y $(u, w) \in E^-$.
- (ii). no existen $v, u \in V$ con $\sigma(v) > \sigma(u) > \sigma(w)$, tales que $(v, w) \in E^+$ y $(u, w) \in E^-$.

Lema 2.44. Sea $G = (V, E^+ \cup E^-)$ un grafo signado y σ un ordenamiento sobre V . El orden dado por σ genera una representación *friend-enemy* de $G \Leftrightarrow \sigma$ es un ordenamiento paraguas de G^+ .

Demostración.

- \Leftarrow) Supongamos que σ no generara una representación *friend-enemy* de G . Entonces existen $v, u, w \in V$ tales que $\sigma(v) < \sigma(u) < \sigma(w)$ con $(v, w) \in E^+$ y $(v, u) \in E^-$. Pero como $(v, w) \in E^+$ y σ es un ordenamiento paraguas de G^+ , debe ocurrir $(v, u) \in E^+$, con lo que se llega al absurdo.
- \Rightarrow) Supongamos que σ no fuera un ordenamiento paraguas, luego existen $v, u, w \in V$ con $\sigma(v) < \sigma(u) < \sigma(w)$ tales que $(v, w) \in E^+$ y $((v, u) \notin E^+ \text{ ó } (u, w) \notin E^+)$. Si sucediera que $(v, u) \notin E^+$, como G es completo, $(v, u) \in E^-$. Si sucediera que $(u, w) \notin E^+$, por la misma razón valdría $(u, w) \in E^-$. En ambos casos se llega a un absurdo pues no se cumple la propiedad *friend-enemy*.

□

Corolario 2.45. Un grafo signado $G = (V, E^+ \cup E^-)$ tiene representación *friend-enemy* en $\mathbb{R} \Leftrightarrow G^+$ es un grafo de intervalos propio.

Demostración. Se deduce por el Corolario 2.42 y el Lema 2.44.

□

2.4. Grafos signados genéricos

Una de las preguntas que surgen al determinar si un grafo tiene representación *friend-enemy* sobre \mathbb{R} , es qué sucede cuando en un grafo signado se permite la inexistencia de aristas entre sus vértices. ¿El problema de reconocimiento sigue siendo polinomial? Esta pregunta fue sugerida como trabajo futuro por Kermarrec y Thraves en [14] y abordada y resuelta por Cygan et. al. en [4]: el problema es NP-Completo. En la primera sección analizaremos y demostraremos estos resultados. Luego analizaremos cotas inferiores de complejidad temporal para resolver el problema. Por último analizaremos una cota superior, fijada por un algoritmo exponencial que lo resuelve.

2.4.1. Análisis de NP-Compleitud

En primer lugar definiremos el problema en cuestión como LINE CLUSTER EMBEDDING en función de lo que presenta el Lema 2.43.

LINE CLUSTER EMBEDDING

Instancia: Un grafo signado $G = (V, E^+ \cup E^-)$.

Pregunta: ¿Existe un ordenamiento σ sobre V tal que para todo $w \in V$ se cumpla que
(i) no existen $v, u \in V$ con $\sigma(v) < \sigma(u) < \sigma(w)$, tales que $(v, w) \in E^+$ y $(u, w) \in E^-$;
(ii) no existen $v, u \in V$ con $\sigma(v) > \sigma(u) > \sigma(w)$, tales que $(v, w) \in E^+$ y $(u, w) \in E^-$?

Para probar el resultado, Cygan et. al. utilizan el problema ACYCLIC PARTITION, que se sabe que es NP-Completo (tanto por Eppstein y Mumford [5] como por Guillemot [8]).

ACYCLIC PARTITION

Instancia: Un grafo dirigido $G = (V, E)$.

Pregunta: ¿Es posible particionar a V en dos conjuntos V_1, V_2 de forma que los subgrafos inducidos $G[V_1]$ y $G[V_2]$ sean grafos dirigidos acíclicos (DAGs)?

Una manera de probar que el anterior problema es NP-Completo es basándose en el problema SET SPLITTING, el cual se sabe que es NP-Completo [6].

SET SPLITTING

Instancia: Una familia de subconjuntos \mathcal{F} de un conjunto finito U , donde $\mathcal{F} \subseteq 2^U$.

Pregunta: ¿Existe una partición de U en dos conjuntos U_1, U_2 tal que ningún conjunto de \mathcal{F} sea contenido completamente por U_1 o por U_2 ?

Para demostrar que un problema π es NP-Completo, deben probarse dos cuestiones:

- $\pi \in \text{NP}$, o lo que es equivalente: dadas una instancia que responde *sí* a la pregunta de π y evidencia (o certificado; de tamaño polinomial en función de la entrada) de la misma, se puede verificar la respuesta *sí* en tiempo polinomial.
- $\forall \pi' \in \text{NP}$, $\pi' \leq_p \pi$ (π es NP-Difícil). En este caso, $\pi' \leq_p \pi$ significa que se puede reducir en tiempo polinomial una instancia de π' en una de π . Por lo cual, basta seleccionar un problema π' que sepamos que es NP-Difícil (es decir, que $\forall \pi'' \in \text{NP}$,

$\pi'' \leq_p \pi'$ y si sólo probamos que $\pi' \leq_p \pi$, por transitividad obtendríamos que $\forall \pi'' \in NP, \pi'' \leq_p \pi$.

Lema 2.46. *El problema ACYCLIC PARTITION es NP-Completo. Más aún, existe un algoritmo de complejidad temporal polinomial que dada una instancia (\mathcal{F}, U) para SET SPLITTING devuelve una instancia equivalente $G = (V, E)$ de ACYCLIC PARTITION, para la cual se cumplen $|V| = |U| + \sum_{F \in \mathcal{F}} |F|$ y $|E| = 3 \sum_{F \in \mathcal{F}} |F|$.*

Demostración. Sean $\pi_1 = \text{ACYCLIC PARTITION}$ y $\pi_2 = \text{SET SPLITTING}$.

En primer lugar veremos que π_1 es NP:

Si nos dan un grafo dirigido $G = (V, E)$ y queremos corroborar si es un DAG, podemos simplemente ejecutar el algoritmo DFS desde cada vértice y verificar si en algún momento se llega a un vértice ya marcado por el algoritmo: si así fuera no se trataría de un DAG, de lo contrario sí lo sería. Claramente es polinomial, con complejidad $O(n(n+m))$ ¹.

Veamos la siguiente transformación polinomial f de π_2 en π_1 :

Crearemos en primer lugar el grafo dirigido $G = (V, E)$ de la siguiente forma: para cada conjunto $F \in \mathcal{F}$ y cada vértice $v \in F$, crearemos un vértice c_v^F y conectaremos todos los vértices $c_{v_i}^F$ del mismo conjunto F (con $1 \leq i \leq |F|$) de forma que generen un ciclo dirigido (inducido en $G[F]$) en cualquier orden. Luego, para cada $u \in U$, crearemos un vértice d_u y para cada vértice de la forma c_u^F , agregaremos los arcos (d_u, c_u^F) y (c_u^F, d_u) a E . Es trivial ver que $|V| = |U| + \sum_{F \in \mathcal{F}} |F|$ y $|E| = 3 \sum_{F \in \mathcal{F}} |F|$.

Dada una instancia \mathcal{I} de π_2 , veamos que $\pi_2(\mathcal{I})$ es *sí* $\Leftrightarrow \pi_1(f(\mathcal{I}))$ es *sí*.

\Leftarrow) Sean V_1, V_2 una solución de π_1 sobre $f(\mathcal{I}) = f(\mathcal{F}, U)$. Consideremos $U_1 = \{u : d_u \in V_1\} \subseteq U$ y $U_2 = U \setminus U_1 = \{u : d_u \in V_2\}$. Veamos que $\{U_1, U_2\}$ son una solución para π_2 sobre \mathcal{I} . Por construcción, hay ciclos C_2 entre d_u y c_u^F , pero como $\{V_1, V_2\}$ fueron solución de π_1 , debe ocurrir que estén separados en distintos conjuntos; luego si $F \in \mathcal{F}$, para todo $u \in U_1$ ocurre que $c_u^F \in V_2$ y para todo $u \in U_2$, $c_u^F \in V_1$.

Sea $F \in \mathcal{F}$. Sabemos que el ciclo que formaban todos los c_u^F no es un ciclo inducido en los subgrafos $G[V_1]$ ni en $G[V_2]$, entonces deben existir $u_1, u_2 \in F$ tales que $c_{u_1}^F \in V_1$ y $c_{u_2}^F \in V_2$. Como los ciclos formados por $\{d_{u_1}, c_{u_1}^F\}$ y $\{d_{u_2}, c_{u_2}^F\}$ no están contenidos totalmente en $G[V_1]$ ni en $G[V_2]$, debe ocurrir que $d_{u_1} \in V_2$ y $d_{u_2} \in V_1$. Por lo cual $u_1 \in U_2$ y $u_2 \in U_1$ y como $U_2 = U \setminus U_1$, resulta que F no está totalmente contenido ni por U_1 ni por U_2 .

\Rightarrow) Sean U_1, U_2 una solución de π_2 sobre $\mathcal{I} = (\mathcal{F}, U)$. Consideremos a $V_1 = \{d_u : u \in U_1\} \cup \{c_u^F : F \in \mathcal{F}, u \in U_2\}$ y $V_2 = \{d_u : u \in U_2\} \cup \{c_u^F : F \in \mathcal{F}, u \in U_1\}$. Tanto en V_1 como en V_2 sucede que no contienen vértices tales que para un mismo F generen un ciclo dirigido entre sus c_u^F , dado que como $\pi_2(\mathcal{I})$ es *sí*, ningún $F \in \mathcal{F}$ está contenido completamente ni en U_1 ni en U_2 , por lo tanto todo ciclo quedó cortado.

Por otro lado, para todo $u \in U$ sucede que d_u queda aislado en el subgrafo inducido $G[U_i]$, pues todos los c_v^F a quienes se conectaba pertenecen a U_{3-i} (para $1 \leq i \leq 2$).

¹ Alternativamente se puede ejecutar el algoritmo de Kahn [13] para conseguir un orden topológico del DAG y corroborar que no haya ciclos, en $O(n+m)$.

Entonces, $G[V_1]$ y $G[V_2]$ no contienen ciclos sino sólo caminos dirigidos y vértices aislados, por lo cual $\{V_1, V_2\}$ es una solución para π_1 sobre $f(\mathcal{I})$.

□

Teorema 2.47. *El problema LINE CLUSTER EMBEDDING es NP-Completo. Más aún, existe un algoritmo de complejidad temporal polinomial que dada una instancia de un grafo dirigido $G = (V_1, E_1)$ para ACYCLIC PARTITION devuelve una instancia equivalente $H = (V_2, E_2^+ \cup E_2^-)$ para LINE CLUSTER EMBEDDING, para la cual se cumplen $|V_2| = |V_1| + |E_1| + 1$, $|E_2^+| = 2|E_1|$ y $|E_2^-| = |E_1| + |V_1|$.*

Demostración. Sean $\pi_1 = \text{LINE CLUSTER EMBEDDING}$ y $\pi_2 = \text{ACYCLIC PARTITION}$.

En primer lugar veremos que π_1 es NP:

Dado un ordenamiento σ de V , de forma que $\sigma(v_1) < \dots < \sigma(v_n)$, simplemente podemos chequear para cada v_i los vértices que tiene a su izquierda en orden y verificando que luego de un v_j ($j < i$) tal que $(v_j, v_i) \in E^-$, no haya más a la izquierda un v_k ($k < j$) tal que $(v_k, v_i) \in E^+$. Análogamente, se puede aplicar la idea a la derecha. Dado que es un chequeo lineal desde cada v_i , la complejidad resulta $O(n^2)$.

Veamos la siguiente transformación polinomial f de π_2 en π_1 :

Dado un grafo dirigido $G = (V_1, E_1)$, crearemos el grafo signado $H = (V_2, E_2^+ \cup E_2^-)$ de la siguiente forma.

El conjunto V_2 consistirá en:

- un vértice distinguido s .
- por cada $e \in E_1$, un vértice de chequeo c_e .
- por cada $v \in V_1$, un vértice de alineamiento a_v .

Por otro lado, se construyen E_2^+, E_2^- como:

- por cada $e \in E_1$, se agrega (s, c_e) en E_2^+ .
- por cada $v \in V_1$, se agrega (s, a_v) en E_2^- .
- por cada arista $(v, w) \in E_1$, se agrega $(c_{(v,w)}, a_v)$ en E_2^+ y $(c_{(v,w)}, a_w)$ en E_2^- .

Es trivial ver que se cumplen $|V_2| = |V_1| + |E_1| + 1$, $|E_2^+| = 2|E_1|$ y $|E_2^-| = |E_1| + |V_1|$.

Dada una instancia \mathcal{I} de π_2 , veamos que $\pi_2(\mathcal{I})$ es sí $\Leftrightarrow \pi_1(f(\mathcal{I}))$ es sí.

\Leftarrow) Sea σ una solución de π_1 sobre $f(\mathcal{I}) = f(V_1, E_1)$. Dado que σ cumple los puntos (i) y (ii) del Lema 2.43 (por ser solución) y s está conectado con aristas positivas a todos los vértices de chequeo y con aristas negativas a todos los vértices de alineamiento, debe ocurrir que $\{s, c_{e_1}, c_{e_2}, \dots, c_{e_{|E_1|}}\}$ estén todos consecutivos en el ordenamiento σ . Es decir, formarán un bloque de vértices juntos y luego a la izquierda y a la derecha estarán los a_v , en algún orden entre ellos. Llamemos $W_1 = \{v \in V_1 : \sigma(a_v) \leq \sigma(s)\}$ y $W_2 = \{v \in V_1 : \sigma(s) \leq \sigma(a_v)\}$ al conjunto de los a_v que están respectivamente a izquierda y a derecha de dicho bloque, y veamos que $\{W_1, W_2\}$ es una solución de π_2 .

Dada cualquier arista (v, u) tal que $v, u \in W_1$, ocurre simultáneamente que $\sigma(a_v) < \sigma(c_{(v,u)})$, $\sigma(a_u) < \sigma(c_{(v,u)})$, $(c_{(v,u)}, a_v) \in E^+$ y $(c_{(v,u)}, a_u) \in E^-$; por lo cual debe ocurrir $\sigma(a_u) < \sigma(a_v)$. Supongamos que $G[W_1]$ tuviera un ciclo dirigido C_k con vértices a_{v_1}, \dots, a_{v_k} , luego por lo visto anteriormente debe ocurrir $\sigma(a_{v_k}) < \dots < \sigma(a_{v_1})$ y como se trata de un ciclo, $(a_{v_k}, a_{v_1}) \in W_1$ y $\sigma(a_{v_1}) < \sigma(a_{v_k})$, lo cual conduce a un absurdo y por lo tanto $G[W_1]$ debe ser acíclico. Análogamente se llega a que $G[W_2]$ debe ser acíclico, por lo cual $\{W_1, W_2\}$ es una solución de π_2 .

\Rightarrow) Sea $\{W_1, W_2\}$ una solución de π_2 sobre $\mathcal{I} = (V_1, E_1)$, luego tanto $G[W_1]$ como $G[W_2]$ tienen órdenes topológicos, llamémoslos σ_1 y σ_2 respectivamente. Sabemos que si $(v, u) \in E_{G[W_1]}$, luego $\sigma_1(v) < \sigma_1(u)$. De la misma forma ocurre en $G[W_2]$ con σ_2 . Veremos que existe un ordenamiento σ que es solución de π_1 para la instancia $(V_2, E_2^+ \cup E_2^-)$ y lo construiremos de la siguiente forma, de izquierda a derecha (como se ve en la Figura 2.3):

- (i) Primero irán todos los vértices a_v para cada $v \in W_1$. Si el orden que tenían dado por σ_1 era $\sigma_1(a_{v_1}) < \dots < \sigma_1(a_{v_{|W_1|}})$, entonces en σ irán invertidos: $\sigma_1(a_{v_{|W_1|}}) < \dots < \sigma_1(a_{v_1})$.
- (ii) Luego irán todos los vértices de chequeo $c_{(v,u)}$ para los cuales $v \in W_1$ y $u \in W_2$. Mientras estén todos juntos, entre ellos puede haber cualquier orden.
- (iii) Luego irán todos los vértices de chequeo $c_{(v,u)}$ para los cuales $v, u \in W_1$. En este caso el orden de los mismos será el orden lexicográfico *inverso* dado por σ_1 sobre los pares (v, u) . Es decir, $\sigma_1((v_1, u_1)) > \sigma_1((v_2, u_2)) \Leftrightarrow (\sigma_1(v_1) < \sigma_1(v_2))$ ó $(\sigma_1(v_1) = \sigma_1(v_2) \text{ y } \sigma_1(u_1) < \sigma_1(u_2))$.
- (iv) Luego irá el vértice distinguido s .
- (v) Luego irán todos los vértices de chequeo $c_{(v,u)}$ para los cuales $v, u \in W_2$. En este caso el orden de los mismos será el orden lexicográfico dado por σ_2 sobre los pares (v, u) .
- (vi) Luego irán todos los vértices de chequeo $c_{(v,u)}$ para los cuales $v \in W_2$ y $u \in W_1$. Mientras estén todos juntos, entre ellos puede haber cualquier orden.
- (vii) Por último irán todos los vértices a_v para cada $v \in W_2$ en el orden que tenían dado por σ_2 : $\sigma_2(a_{v_1}) < \dots < \sigma_2(a_{v_{|W_2|}})$.

Veamos que este orden σ es solución de π_1 . Llamaremos *condición* a las condiciones (i) y (ii) del Lema 2.43 (que es lo que busca cumplir π_1).

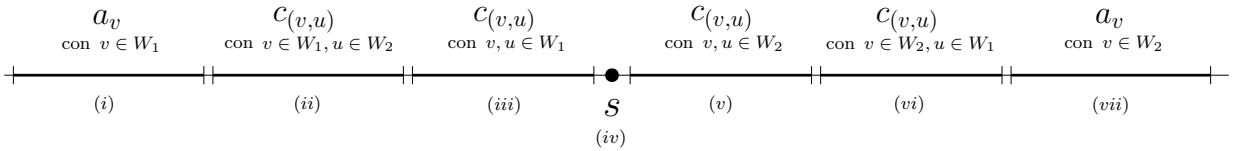


Fig. 2.3: Orden σ construido a partir de σ_1 y σ_2 .

En primer lugar s cumple la *condición*, pues tiene a su alrededor todos los vértices de chequeo (con los cuales tiene aristas positivas), y por fuera de ese bloque que lo rodea tiene los vértices de alineamiento (con los cuales tiene aristas negativas).

Luego, sea un vértice de chequeo $c_{(v,u)}$.

- Si $v \in W_1, u \in W_2$, $c_{(v,u)}$ está en la sección dada por (ii) y se conecta con una arista positiva a a_v y una negativa a a_u . Como $\sigma(a_v) < \sigma(a_u)$, es claro que se cumple la *condición*. El caso es análogo si $v \in W_2, u \in W_1$, en cuyo caso $c_{(v,u)}$ provendrá de la sección dada por (vi).
- Si $v, u \in W_1$, se une con una arista positiva a a_v y con una negativa a a_u . Por otro lado, como σ_1 es un orden topológico en $G[W_1]$, $\sigma_1(v) < \sigma_1(u)$, que por (i) ocurre que $\sigma(u) < \sigma(v)$. Luego se cumple la *condición* pues $c_{(v,u)}$ está en la sección dada por (iii). El caso es análogo si $v, u \in W_2$, pero usando $c_{(v,u)}$ de la sección dada por (v) y el orden de a_v, a_u dado por (vii).

Además, $c_{(v,u)}$ es indiferente hacia los demás c_e y a_w con $w \notin \{v, u\}$, por lo cual cumple la *condición*.

Luego, sea un vértice de alineación $a_v \in W_1$ (cuando $a_v \in W_2$, la demostración es análoga). Los vecinos de a_v pueden ser:

1. vecinos $c_{(v,u)}$ unidos por aristas positivas, con $u \in W_2$.
2. vecinos $c_{(v,u)}$ unidos por aristas positivas, con $u \in W_1$.
3. vecinos $c_{(u,v)}$ unidos por aristas negativas, con $u \in W_1$.
4. vecinos $c_{(u,v)}$ unidos por aristas negativas, con $u \in W_2$.

Por un lado, el orden que tienen los vértices de chequeo sobre σ garantiza que los vecinos de (1) queden ubicados antes que los vecinos de (2), pues todos los $c_{(v,u_1)}$ del punto (ii) están a izquierda de todos los $c_{(v,u_2)}$ del punto (iii), con $u_1 \in W_1$ y $u_2 \in W_2$. Además, σ garantiza que los vecinos de (3) queden ubicados antes que los de (4), pues todos los $c_{(u_1,v)}$ del punto (ii) están a izquierda de todos los $c_{(u_2,v)}$ del punto (vi), con $u_1 \in W_1$ y $u_2 \in W_2$. Por otro lado, asumamos por absurdo que los vértices de (2) no fueran antes que los de (3): entonces existen u_1, u_2 tales que $(v, u_1), (u_2, v) \in E_1$ pero $\sigma(c_{(u_2,v)}) > \sigma(c_{(v,u_1)})$. Entonces, como σ_1 era un orden topológico de $G[W_1]$ debe ocurrir $\sigma_1(u_2) < \sigma_1(v)$, pero luego debería valer $\sigma(c_{(u_2,v)}) < \sigma(c_{(v,u_1)})$ pues en (iii) ocurre que los vértices fueron ubicados en orden lexicográfico *inverso*. Por esto, se llega a un absurdo, y los vecinos de (2) deben ir antes que los de (3) en σ . Entonces, dado que (1) < (2) < (3) < (4) y que a_v se conecta con aristas negativas con todos los vértices a partir de s hasta el final (dado σ), se cumple la *condición* sobre a_v .

Luego σ debe ser solución de π_1 para $f(\mathcal{I})$.

Finalmente, uniendo el resultado obtenido junto con el Lema 2.46, concluimos que el problema LINE CLUSTER EMBEDDING es NP-Completo. □

Corolario 2.48. Es NP-Difícil decidir la menor dimensión sobre \mathbb{R} (con distancia Euclídea) sobre la cual un grafo signado genérico puede tener representación *friend-enemy*.

Para finalizar con esta sección, el Teorema 2.37 nos dice que saber si un grafo signado genérico tiene representación *friend-enemy* sobre \mathbb{R} es un problema NP-Completo. De ésta misma razón, se deriva que es NP-Difícil decidir la menor dimensión en la cual existe esta representación (Corolario 2.38).

2.4.2. Cota inferior sobre la complejidad temporal

Las reducciones realizadas en la sección anterior nos permiten encontrar cotas inferiores sobre la complejidad temporal para resolver el problema LINE CLUSTER EMBEDDING, haciendo uso de la equivalencia que tiene con los problemas ACYCLIC PARTITION y SET SPLITTING. Analizaremos en esta sección el aporte de Cygan et. al. sobre la posibilidad de reducir SET SPLITTING a un problema llamado 3-CNF-SAT, sobre el cual Impagliazzo y Paturi (haciendo uso del *Exponential Time Hypothesis*) hallaron una cota inferior del orden $O^*(2^{\epsilon n})$ en [11].

El problema SAT (o *problema de la satisfacibilidad booleana*) se basa en decidir si existe una interpretación para la cual se satisface una fórmula booleana. Cualquier fórmula puede ser representada como una conjunción de cláusulas (fórmulas más pequeñas), en donde cada una es una disyunción de literales (variables); por lo cual se puede representar a F de la forma $F = F_1 \wedge F_2 \wedge \dots \wedge F_n$ donde $F_i = p_{i,1} \vee p_{i,2} \vee \dots \vee p_{i,m_i}$. Esta forma se llama CNF (o *forma normal conjuntiva*).

El problema general k -CNF-SAT es un derivado del anterior y se pregunta si existe una asignación de valores booleanos a variables para que se satisfaga una fórmula en CNF con cláusulas de k literales.

3-CNF-SAT

Instancia: Una fórmula booleana \mathcal{F} expresada en CNF, con cláusulas de 3 literales.

Pregunta: ¿Existe una signación de valores booleanos a las variables de \mathcal{F} para que la fórmula se satisfaga?

Lema 2.49. *Existe un algoritmo de complejidad temporal polinomial que dada una instancia φ para 3-CNF-SAT con n variables y m cláusulas, devuelve una instancia equivalente (\mathcal{F}, U) para SET SPLITTING, para la cual se cumplen $\sum_{F \in \mathcal{F}} |F| = 2n + 4m$ y $|U| = 2n + 1$.*

Demostración. Sean $\pi_1 = \text{SET SPLITTING}$ y $\pi_2 = k\text{-CNF-SAT}$.

Veamos la siguiente transformación polinomial f de π_2 en π_1 :

Crearemos (\mathcal{F}, U) de la siguiente forma: por un lado, U contendrá un elemento distinguido s y por cada variable x en φ , tendrá a los literales x y $\neg x$. Por otro lado, \mathcal{F} tendrá un conjunto $F_x = \{x, \neg x\}$ por cada variable x en φ y además, por cada cláusula C de φ , un conjunto F_C que contendrá todos los literales de C y el elemento distinguido s .

Se puede constatar que $\sum_{F \in \mathcal{F}} |F| = 2n + 4m$ y $|U| = 2n + 1$.

Dada una instancia \mathcal{I} de π_2 , veamos que $\pi_2(\mathcal{I})$ es *sí* $\Leftrightarrow \pi_1(f(\mathcal{I}))$ es *sí*.

\Leftarrow) Sea ψ un conjunto de valuaciones booleanas de las variables de φ que resulta solución de π_2 para $\mathcal{I} = \varphi$ (es decir, satisfacen φ). Construiremos un conjunto $X \subseteq U$ en donde X contiene todos los literales que valen *true* en ψ .

Por un lado, ningún $F_x \in \mathcal{F}$ es contenido por X ni por $U \setminus X$ pues $F_x = \{x, \neg x\}$ y tanto X como $U \setminus X$ sólo pueden tener uno de esos elementos. Por otro lado, ningún $F_C \in \mathcal{F}$ puede ser contenido por X ni por $U \setminus X$ pues como φ es satisfecha por

ψ , todas las cláusulas C son *true* y deben tener al menos un literal $\alpha_{i,C}$ con valor *true* pues se trata de una disyunción de literales. Luego $\alpha_{i,C} \in X$ pero $s \notin X$, y $\alpha_{i,C} \notin U \setminus X$ pero $s \in U \setminus X$.

\Rightarrow) Sea $\{X, U \setminus X\}$ una solución de π_1 para $f(\mathcal{I}) = f(\varphi)$. Supongamos que $s \notin X$, entonces como ningún $F_x \in \mathcal{F}$ es contenido por X , sucede que para cada variable x , ocurre $x \in X$ ó $\neg x \in X$ y no pueden darse ambas condiciones a la vez. Luego X tiene un literal de x por cada variable x . Consideremos ψ como el conjunto de valuaciones booleanas que validan X . Entonces ψ satisface φ pues para todo $F_C \in \mathcal{F}$, X debe tener un literal que comparta con F_C porque como $s \notin X$, de lo contrario ocurriría que F_C estaría contenido en X . Si $s \in X$, simplemente aplicamos el mismo razonamiento pero con $U \setminus X$ y obtenemos un ψ que satisface φ .

□

El último paso a realizar es analizar la cota mínima de complejidad temporal de LINE CLUSTER EMBEDDING, teniendo en cuenta las equivalencias de los Teoremas 2.46, 2.47 y el Lema 2.49 y realizando el cálculo del tamaño de las instancias a medida que se transforman. Para llegar al número que encontraron Cygal et. al. se supone válida la *Exponential Time Hypothesis*, que sostiene que existe una constante $\alpha > 0$ para la cual no existe un algoritmo que resuelva el problema 3-CNF-SAT en tiempo $O(2^{\alpha n})$, donde n indica la cantidad de variables libres de la fórmula que se quiere satisfacer.

Por otro lado, se necesita hacer uso del *Sparsification Lemma*, probado por Impagliazzo et. al. en [12]. El lema dice lo siguiente.

Lema 2.50. *Para todo $\epsilon > 0$ y un k positivo, existe una constante α tal que cualquier fórmula Φ para k -SAT con n variables puede expresarse como $\Phi = \bigvee_{i=1}^t \Psi_i$, con $t \leq 2^{\alpha n}$ y cada Ψ_i es una fórmula para k -SAT con a lo sumo αn cláusulas. Más aún, esta disyunción puede ser computada por un algoritmo de complejidad temporal $O^*(2^{\epsilon n})$.*

Notación 4. La notación $O^*(\cdot)$ no tiene en cuenta factores que son polinomiales en el tamaño de la entrada.

Teorema 2.51. *Si el Exponential Time Hypothesis es cierto, entonces existe una constante $\delta > 0$ para la cual no existe un algoritmo que al tomar una instancia $(V, E^+ \cup E^-)$ de LINE CLUSTER EMBEDDING lo resuelva en tiempo $O(2^{\delta(|V|+|E^+|+|E^-|)})$.*

Demostración. Supongamos por el absurdo que para todo $\delta > 0$ existe un algoritmo que puede resolver el LINE CLUSTER EMBEDDING en $O(2^{\delta(|V|+|E^+|+|E^-|)})$. La idea, bajo esta suposición, es construir un algoritmo que resuelva 3-CNF-SAT en tiempo $O^*(2^{\epsilon n})$ para todo $\epsilon > 0$, con n el número de variables libres. De esta forma, habría una contradicción con el *Exponential Time Hypothesis* y se llegaría al absurdo.

Dado un $\epsilon > 0$ fijo, realizamos lo siguiente:

- Primero tomar una instancia φ para 3-CNF-SAT y por el Lema 2.50 expresarlo como una disyunción Ψ de a lo sumo $2^{\epsilon n/2}$ instancias para 3-CNF-SAT, cada una conteniendo como mucho αn cláusulas (para alguna constante α). Esto se hará en $O^*(2^{\epsilon n/2})$.
- Luego reducir cada instancia en tiempo polinomial aplicando las transformaciones a Set Splitting y a ACYCLIC PARTITION, para conseguir a lo sumo $2^{\epsilon n/2}$ instancias de LINE CLUSTER EMBEDDING, cada una con $|V|, |E^+|, |E^-| \leq \beta n$ (para alguna constante β).
- Finalmente, para cada una de las instancias, correr el algoritmo para LINE CLUSTER EMBEDDING. Esto cuesta $O(2^{\delta(|V|+|E^+|+|E^-|)})$ con $\delta = \frac{\epsilon}{6\beta}$.

□

Para finalizar con esta sección, el Teorema 2.41 nos dice que si asumimos válido el *Exponential Time Hypothesis*, saber si un grafo signado genérico $G = (V, E^+ \cup E^-)$ es *friend-enemy* tiene una cota mínima de complejidad temporal $O(2^{\delta(|V|+|E^+|+|E^-|)})$, para alguna constante $\delta > 0$.

2.4.3. Algoritmo exponencial para el Line Cluster Embedding

En esta última sección destinada al caso general de los grafos signados *friend-enemy* sobre \mathbb{R} , se cierra bastante el esquema al analizar un algoritmo exponencial que resuelva el problema de forma genérica. Un algoritmo de fuerza bruta podría hallar y chequear todos los órdenes de vértices de G en $O^*(n!)$. Pero la solución que propusieron Cygal et. al. en [4] es la de un algoritmo que utiliza programación dinámica y funciona en un tiempo exponencial aceptable.

Antes de pasar al algoritmo, veamos la siguiente definición y lema.

Definición 2.52. Sea $G = (V, E^+ \cup E^-)$ un grafo signado. Dados $X \subseteq V$ y $v \in V \setminus X$, decimos que v es *bueno* para X si y solo si:

1. $\nexists x \in X$ tal que $(v, x) \in E^-$ y $(u, x) \in E^+$, para algún $u \in V \setminus (X \cup \{v\})$.

Es decir, no existe un vértice en X que se conecte de forma negativa con algún v y al mismo tiempo de forma negativa con alguien de $V \setminus (X \cup \{v\})$.

2. $\nexists x \in V \setminus (X \cup \{v\})$ tal que $(v, x) \in E^-$ y $(u, x) \in E^+$, para algún $u \in X$.

Es decir, no existe un vértice en $V \setminus (X \cup \{v\})$ que se conecte de forma negativa con algún v y al mismo tiempo de forma negativa con alguien de X .

Lema 2.53. Sea $G = (V, E^+ \cup E^-)$ un grafo signado. Un ordenamiento σ es una solución de LINE CLUSTER EMBEDDING para $G \Leftrightarrow$ para todo $v \in V$, v resulta bueno para el conjunto $\{u : \sigma(u) < \sigma(v)\}$.

Demostración.

\Leftarrow) Todo $v \in V$ resulta bueno para $X = \{u : \sigma(u) < \sigma(v)\}$, entonces sea $v \in V$ un vértice cualquiera. Supongamos que σ no fuera solución de LINE CLUSTER EMBEDDING: entonces una opción es que existan dos vértices $u_1, u_2 \in V$ tales que $\sigma(u_1) < \sigma(u_2) < \sigma(v)$ con $(u_1, v) \in E^+$ y $(u_2, v) \in E^-$. En ese caso u_2 no resultaría *bueno* para $\{u : \sigma(u) < \sigma(u_2)\}$, lo cual sería absurdo pues por hipótesis todo $w \in V$ era *bueno*. La opción restante es que existan dos vértices $u_1, u_2 \in V$ tales que $\sigma(u_1) > \sigma(u_2) > \sigma(v)$ con $(u_1, v) \in E^+$ y $(u_2, v) \in E^-$. Pero nuevamente u_2 no resultaría *bueno* para $\{u : \sigma(u) < \sigma(u_2)\}$ por la misma razón.

Por lo cual, σ debe ser solución de LINE CLUSTER EMBEDDING.

\Rightarrow) Supongamos que v no fuera *bueno* para $X = \{u : \sigma(u) < \sigma(v)\}$. Entonces deben existir vértices $x \in X$, $u \in V \setminus (X \cup \{v\})$, con $(x, u) \in E^+$ que invaliden alguno de los puntos de la Definición 2.52. En particular, como $x \in X$, ocurre que $\sigma(x) < \sigma(v)$, y como $u \notin X$, ocurre que $\sigma(v) < \sigma(u)$. Luego analizando los mencionados puntos, en cada uno ocurre que:

1. $(v, x) \in E^-$, por lo cual $\sigma(x) < \sigma(v) < \sigma(u)$ con $(x, u) \in E^+$ y $(x, v) \in E^-$, que se contradice con que σ sea solución de LINE CLUSTER EMBEDDING.
2. $(v, u) \in E^-$, por lo cual $\sigma(x) < \sigma(v) < \sigma(u)$ con $(u, x) \in E^+$ y $(u, v) \in E^-$, que se contradice con lo mismo que el punto anterior

Por lo tanto no puede ocurrir que exista un v *no bueno* para X y por lo tanto la implicación resulta válida.

□

El Lema 2.53 nos permite obtener el siguiente teorema.

Teorema 2.54. *El problema LINE CLUSTER EMBEDDING puede ser resuelto en costo temporal y espacial $O^*(2^n)$. Más aún, de existir un ordenamiento de vértices que cumpla con el problema, lo devuelve.*

Demostración. Dada una instancia $G = (V, E^+ \cup E^-)$ para LINE CLUSTER EMBEDDING, construiremos el conjunto de pares $W = \{(v, X) : v \text{ es } \textit{bueno} \text{ para } X\}$, para todos los $X \subseteq V$ y $v \in V$. Luego construiremos un grafo dirigido $H = (W, F)$, en donde el conjunto de arcos será dado por $((v, X), (v', X')) \in F \Leftrightarrow X' = X \cup \{v\}$. Usando el Lema 2.53, que exista un ordenamiento σ que resuelva el problema es equivalente a que exista $(v, V \setminus \{v\}) \in W$ tal que pueda ser alcanzado desde algún $(v', \emptyset) \in W$ por medio de un camino dirigido en H . Esto ocurre porque se irían agregando los v en un orden tal que todos los vértices anteriores lo tendrían a v como *bueno* para el conjunto que ellos conforman.

Decidir si este camino dirigido existe y en dicho caso obtenerlo, se puede realizar con BFS en tiempo lineal en la cantidad de vértices y aristas. Por como se construye H , esto cuesta $O^*(2^n)$ pues se descarta el tiempo de reconocer si los pares (v, X) en W cumplen que v sea bueno para X (pues este chequeo es polinomial).

□

Finalmente terminamos esta sección analizando que existe un algoritmo exponencial que resuelve LINE CLUSTER EMBEDDING en $O^*(2^n)$.

3. GRAFOS *FRIEND-ENEMY* SOBRE \mathbb{R}^2

En este capítulo analizaremos primeramente un ejemplo de grafo que no es *friend-enemy* sobre \mathbb{R}^2 , aportado por Kermarrec y Thraves en [15]. Luego de aportar algunas propiedades iniciales, introduciremos la noción de grafo *split* junto con algunas propiedades importantes de la clase. Por último, caracterizaremos por medio de subgrafos prohibidos a los *split friend-enemy* con clique máxima K_2 y presentaremos algunas propiedades que deben cumplir los *split* de clique máxima K_n con $n \geq 3$.

3.1. Propiedades generales

Como primer paso se analizará un subgrafo presentado en [15], el *negative triangle*, el cual no tiene representación *friend-enemy* en \mathbb{R}^2 . Luego se aportarán nuevas propiedades que son de uso general sobre los grafos *friend-enemy* en el plano, con sus respectivas demostraciones.

Definición 3.1. Llamaremos *negative triangle* al grafo que tiene cinco vértices, en donde hay dos ciclos C_4 inducidos con aristas positivas compartiendo tres vértices en común, y en donde las aristas restantes para completar a un K_5 son negativas (Figura 3.1).

Lema 3.2. *El grafo negative triangle no tiene representación friend-enemy en \mathbb{R}^2 .*

Demostración. Supongamos que tuviera una representación *friend-enemy* \mathcal{R} sobre el plano. Llamemos v_0 y v_a al único par de vértices que tienen exactamente una arista negativa (la cual, en particular, comparten entre sí). Dado que en \mathcal{R} v_0 y v_a constituyen una recta, es equivalente rotar la figura y obtener \mathcal{R}' de forma que v_0 quede ubicado en las coordenadas $(0, 0)$ y v_a en $(0, a)$ (para algún valor correspondiente de a); no altera en absoluto la validez de la representación.

Debe ocurrir que los tres vértices adyacentes a v_0 por medio de aristas positivas estén ubicados dentro del círculo de centro v_0 y radio a (pues de lo contrario, v_0 tendría más cerca a su enemigo v_a que a sus tres amigos). De forma análoga debe ocurrir que los mismos tres vértices unidos a v_a por medio de aristas positivas estén ubicados dentro del círculo de centro v_a y radio a . Esto nos dice que los tres vértices a los cuales se conectan v_0 y v_a deben estar en la intersección de estas circunferencias.

La arista negativa entre v_0 y v_a particiona la intersección de circunferencias en dos mitades y como tres vértices deben ser ubicados entre dichas mitades, es claro que deberá haber al menos dos de ellos en una mitad. Dado que espejar la representación no influye en que ésta sea *friend-enemy*, asumimos que a la derecha hay al menos dos vértices v_1 y v_2 , de coordenadas (x_1, y_1) y (x_2, y_2) respectivamente. Tanto x_1 como x_2 poseen valores no negativos por estar a la derecha de v_0 .

Asumimos $x_1 \geq x_2$ (de lo contrario, simplemente se renombran v_1 y v_2). Por otro lado, asumimos $y_1 \leq \frac{a}{2}$ (de lo contrario, simplemente se invierte \mathcal{R}' sobre el eje $y = \frac{a}{2}$). Luego, $d^2(v_1, v_2) = (x_1 - x_2)^2 + (y_1 - y_2)^2$ y $d^2(v_1, v_a) = x_1^2 + (y_1 - a)^2$, por lo que $d^2(v_1, v_a) - d^2(v_1, v_2) = x_1^2 + (y_1 - a)^2 - (x_1 - x_2)^2 - (y_1 - y_2)^2 = 2x_1x_2 - x_2^2 - y_2^2 + 2y_1y_2 + a^2 - 2ay_1 = 2x_1x_2 - x_2^2 + (a - y_2)(y_2 + a - 2y_1) \geq 0$ (esta desigualdad vale pues $x_1 \geq x_2$, lo que hace que $2x_1x_2 - x_2^2 \geq 0$, y pues $a \geq y_2$ y $y_1 \leq \frac{a}{2}$, lo cual fuerza $(a - y_2)(y_2 + a - 2y_1) \geq 0$).

Esta desigualdad lleva a un absurdo pues dice que v_1 está más cerca de v_2 y más lejos de v_a , cuando $(v_1, v_2) \in E^-$ y $(v_1, v_a) \in E^+$. Por lo tanto, el grafo *negative triangle* no puede ser *friend-enemy* sobre el plano. \square

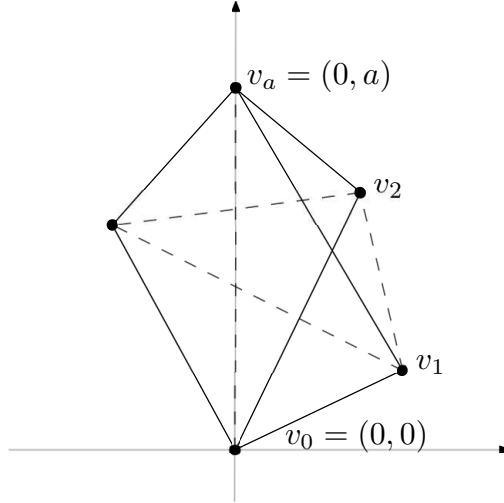


Fig. 3.1: Grafo *negative triangle* representado sobre \mathbb{R}^2

Lema 3.3. *Sea $G = (V, E)$ un grafo con representación *friend-enemy*. Entonces todo subgrafo inducido de G tiene representación *friend-enemy*.*

Demostración. Dado G *friend-enemy*, veremos primero que para todo $v \in V$, $G' = (V \setminus \{v\}, E')$ tiene representación *friend-enemy*.

Consideremos $T = \{(u, w, z) \in V^3 : (u, w) \in E \text{ y } (u, z) \notin E\}$. Como G es *friend-enemy*, sucede que para toda tripla $(u, w, z) \in T$, $d_G(u, w) < d_G(u, z)$ (donde $d_G(v_1, v_2)$ es la distancia Euclídea entre v_1 y v_2 sobre la representación de G).

Dado un $v \in V$ arbitrario, representaremos en el espacio métrico a G' con la misma función de asignación de coordenadas que usaba G , con lo cual G' es exactamente la representación *friend-enemy* de G pero eliminando a v y a todas sus aristas. Por esto, para todos $u, w \in V \setminus \{v\}$, $d_G(u, w) = d_{G'}(u, w)$.

Consideremos ahora a $T' = \{(u, w, z) \in (V \setminus \{v\})^3 : (u, w) \in E' \text{ y } (u, z) \notin E'\}$. Si $(u, w, z) \in T'$, luego ocurre que:

- $(u, w) \in E'$, pero además $(u, w) \in E$ pues haber sacado a v no influyó en la arista entre u y w , por lo cual G' la mantuvo.
- $(u, z) \notin E'$, pero además $(u, z) \notin E$ pues haber sacado a v no influyó en la arista entre u y w , por lo cual G' la mantuvo.

Luego, las condiciones $(u, w) \in E$ y $(u, z) \notin E$ implican que $(u, w, z) \in T$. Por lo tanto, $T' \subseteq T$. Finalmente, como para toda tripla $(u, w, z) \in T'$, $d_G(u, w) < d_G(u, z) \Rightarrow d_{G'}(u, w) < d_{G'}(u, z)$ (para la misma función de asignación de coordenadas), ocurre que

G' tiene representación *friend-enemy*.

De esta forma, partiendo de un grafo G *friend-enemy*, se pueden eliminar uno a uno los vértices necesarios hasta conseguir cualquier subgrafo inducido de G y siempre se conserva la propiedad. □

Corolario 3.4. (Contrarecíproco del Lema 3.3) Si existe un subgrafo inducido de G que no tiene representación *friend-enemy*, entonces G no tiene representación *friend-enemy*.

El Corolario anterior nos muestra la importancia de trabajar con subgrafos prohibidos al caracterizar mediante esta propiedad.

Observación 3.5. Que todo subgrafo inducido propio de G tenga representación *friend-enemy* **no implica** que G sea *friend-enemy*.

Ejemplo 3.6. En relación a la observación anterior, si G fuera un $K_{1,6}$, sus subgrafos inducidos son $K_{1,5}$, son el conjunto independiente de 6 vértices o son subgrafos de ellos; todos ellos con representación *friend-enemy*. Pero como veremos en la sección 3.3.1, los grafos $K_{1,6}$ no cumplen esta propiedad.

Lema 3.7. Sean $G = (V, E)$ un grafo y $v, u, w \in V$ vértices distintos tales que $(v, u), (v, w) \in E^+$ y $(u, w) \notin E^-$. Luego si G admite representación *friend-enemy*, ocurre que $\hat{u}\hat{v}w > 60^\circ$.

Demostración. Asumimos que G tiene representación *friend-enemy* y supongamos por el absurdo que $\alpha = \hat{u}\hat{v}w \leq 60^\circ$. Como se ve en la Figura 3.2, considerando $m = d(v, u)$ no puede suceder que w quede situado dentro de la circunferencia de centro u y radio m (llamada U), pues en ese caso valdría $m = d(u, v) \not\leq d(u, w)$ con $(u, v) \in E^+$ y $(u, w) \in E^-$. Luego w debe quedar situado más allá de U , en el área sombreada.

Supongamos que $\alpha < 60^\circ$. Sabemos que la distancia estrictamente más pequeña entre w y la circunferencia U debe darse sobre la recta entre w y u , por lo cual se puede reescribir $d(w, u) = d(w, U) + m$. Por otro lado, aplicando la misma idea para la recta entre w y v , $d(w, v) = d(w, V) + m$ con $d(w, V) = \varepsilon_1 + \varepsilon_2$, donde ε_2 es el largo desde w hasta el punto más cercano a U y ε_1 es el tamaño del sector de la misma recta entre U y V . Se cumplen $\varepsilon_1 > 0$ (pues $\alpha < 60^\circ$) y $\varepsilon_2 > d(w, U)$ (pues de lo contrario w estaría exactamente sobre el borde de la circunferencia U y luego u tendría a igual distancia a w y a v). Luego $d(w, v) = d(w, V) + m = \varepsilon_1 + \varepsilon_2 + m > \varepsilon_1 + d(w, U) + m > d(w, U) + m = d(w, u)$. Pero entonces $d(w, v) > d(w, u)$ con $(w, v) \in E^+$ y $(w, u) \in E^-$, lo cual lleva al absurdo.

Debe ocurrir entonces que $\alpha = 60^\circ$. Si w estuviera exactamente en el borde de la circunferencia V , no sería una representación *friend-enemy* pues u tendría igual de cerca a su amigo v y a su enemigo w . Por lo cual $d(w, V) > 0$ y esto causa que $\beta = \hat{v}\hat{w}u > 60^\circ$.

Luego, por una propiedad básica de geometría sabemos que en todo triángulo, el mayor lado es opuesto al mayor ángulo. En este caso, como la suma de los ángulos debe dar 180° y $\alpha = 60^\circ$ y $\beta > 60^\circ$, luego $\hat{v}\hat{w}u < 60^\circ$ y por lo tanto $d(w, v) > d(w, u)$. □

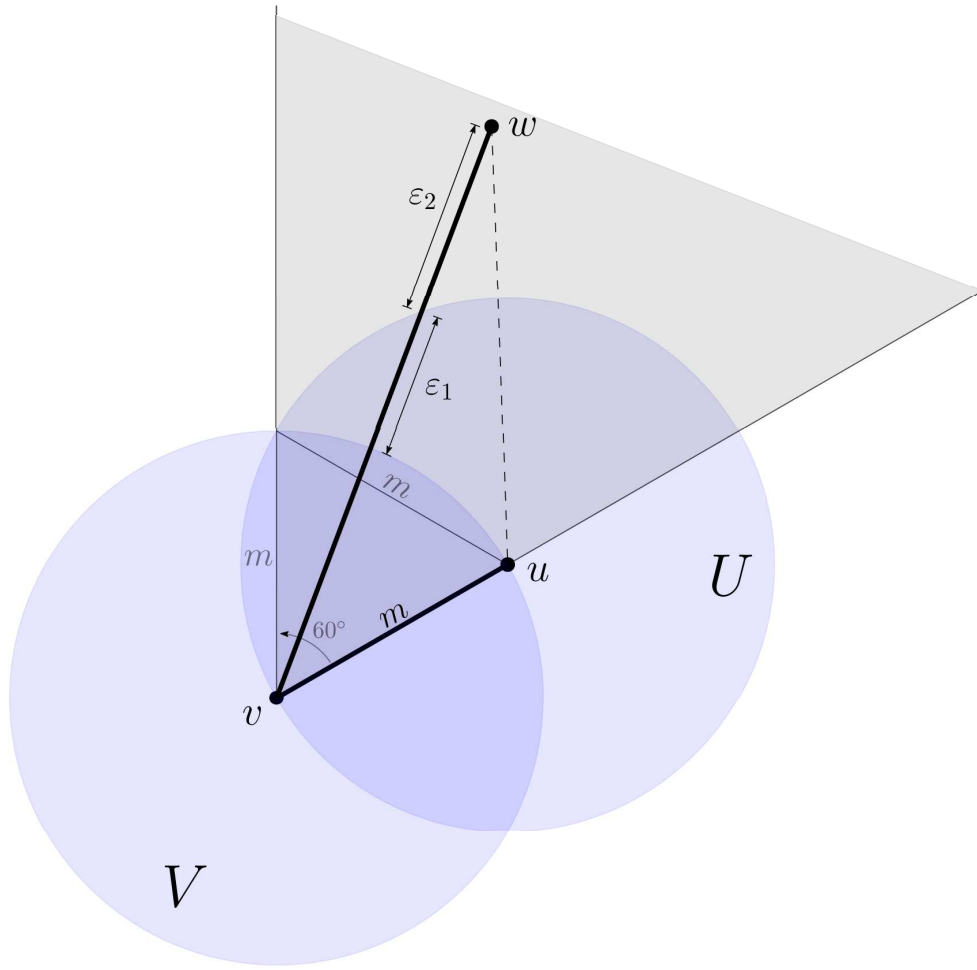


Fig. 3.2: Caso en el que w está ubicado a un ángulo menor o igual a 60° partiendo de u con centro v y donde se lo intenta ubicar por fuera de la circunferencia U .

La demostración del lema anterior lleva a realizar la siguiente definición y lema.

Definición 3.8. Una representación de un grafo $G = (V, E^+ \cup E^-)$ en un espacio métrico (\mathcal{M}, d) se dice *friend-enemy débil* cuando para todos $u, v, w \in V$ tales que $(u, v) \in E^+$ y $(v, w) \in E^-$, se cumple $d(u, v) \leq d(u, w)$.

Lema 3.9. Sean $G = (V, E)$ un grafo y $v, u, w \in V$ vértices distintos tales que $(v, u), (v, w) \in E^+$ y $(u, w) \notin E^-$. Luego si G admite representación friend-enemy débil, ocurre que $\hat{u}vw \geq 60^\circ$.

Demostración. Es análoga a la demostración del Lema 3.7 y en particular puede darse el caso $\hat{u}vw = 60^\circ$ pues esta vez w puede quedar sobre la intersección de los bordes de las circunferencias.

□

3.2. Definiciones y propiedades básicas de grafos *split*

Definición 3.10. Un *conjunto independiente* de un grafo $G = (V, E)$ es un subconjunto $U \subseteq V$ tal que no existen $v_1, v_2 \in U$ tales que $(v_1, v_2) \in E$.

Definición 3.11. Un *clique* de un grafo $G = (V, E)$ es un subconjunto $U \subseteq V$ tal que para todos $v_1, v_2 \in U$, ocurre $(v_1, v_2) \in E$.

Definición 3.12. Un grafo $G = (V, E)$ se dice *split* si puede partitionarse V en dos conjuntos $\{V_1, V_2\}$ tales que $G[V_1]$ es un clique y $G[V_2]$ es un conjunto independiente.

Definición 3.13. Llamaremos a un grafo *split* con partición $\{V_1, V_2\}$ *split completo* si contiene todas las aristas de $V_1 \times V_2$ y V_1 es un completo y V_2 un conjunto independiente.

En las siguientes secciones realizaremos un abuso de notación para no complicar la lectura innecesariamente. Salvo que se especifique lo contrario, cuando se mencione un grafo G , se considerará que G es la parte *positiva* del grafo que realmente se tomará en cuenta (la parte positiva se trata del grafo original sólo con aristas positivas). Además, trabajaremos siempre con grafos signados completos por lo cual, por ejemplo, hablar del *split completo* $K_{1,3}$ representará en realidad al grafo signado completo cuyas aristas positivas forman un $K_{1,3}$ y sus negativas son las aristas restantes para completar a un K_4 .

3.3. Caracterización *friend-enemy* de grafos *split* por subgrafos prohibidos

Una caracterización de una propiedad P sobre una clase de grafos \mathcal{G} por medio de subgrafos prohibidos es un conjunto de grafos $C \subseteq \mathcal{G}$ que garantizan que $G \in \mathcal{G}$ cumple $P \Leftrightarrow G$ no tiene como subgrafo inducido a ningún elemento de C .

Encontramos que particionar a la clase de grafos *split* en función del tamaño del clique máximo es una forma que nos ordena un poco el problema de caracterizar los subgrafos prohibidos. En las siguientes secciones, primero caracterizaremos al conjunto entero de los subgrafos prohibidos para los grafos de clique máximo K_2 y por último aportaremos algunos prohibidos para los de clique máximo K_3 .

3.3.1. Grafos *split* con clique máximo K_2

Comenzaremos analizando los grafos estrella $K_{1,n}$. Veremos, en primer lugar, que el grafo $K_{1,5}$ es *friend-enemy* sobre \mathbb{R}^2 (por lo cual todos los subgrafos inducidos también) y finalmente que ningún $K_{1,n}$ con $n \geq 6$ admite esta representación en el plano.

Lema 3.14. *Dado un pentágono regular, todo lado mide más que la distancia desde el centro a cualquier otro vértice.*

Demostración. Sea un grafo $K_{1,5}$ representado en el plano como un pentágono regular (vértices a, b, c, d, e) con un vértice central f (ver Figura 3.3a). Por ser regular, las distancias al centro son $|(a, f)| = |(b, f)| = |(c, f)| = |(d, f)| = |(e, f)| = m$ y las distancias entre vértices no centrales consecutivos (lados del pentágono) son $|(a, b)| = |(b, c)| = |(c, d)| = |(d, e)| = |(e, a)| = h$.

Luego, $\text{sen}(\frac{\alpha}{2}) = \frac{m}{h/2} \Leftrightarrow \text{sen}(\frac{360^\circ}{5 \cdot 2}) = \frac{2m}{h} \Leftrightarrow h = \frac{2m}{\text{sen}(36^\circ)}$.

Finalmente, como $\frac{2}{\text{sen}(36^\circ)} > 1$, vale $h > m$. □

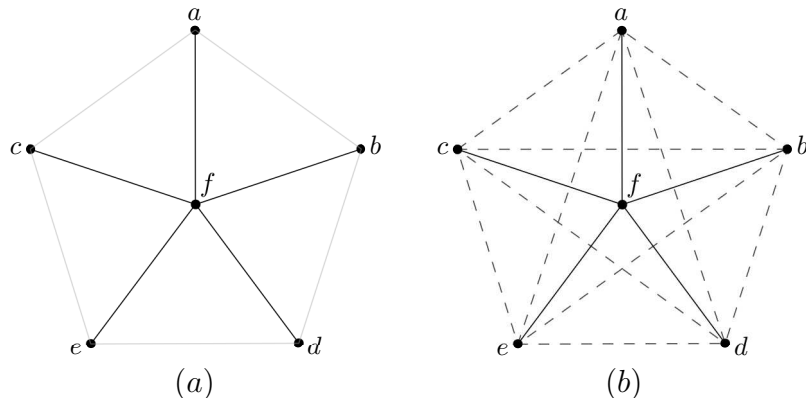


Fig. 3.3: En (a) un grafo $K_{1,5}$ como un pentágono regular y en (b) un grafo signado completo con aristas positivas que conforman un $K_{1,5}$.

Lema 3.15. *El grafo split $K_{1,5}$ tiene representación friend-enemy en \mathbb{R}^2 .*

Demostración. La demostración se basa en representar a G como un pentágono regular con un vértice central (ver Figura 3.3b). Por el Lema 3.13 sabemos que los lados son más pequeños que la distancia del centro a otro vértice. Usamos el mismo nombramiento de vértices y sin pérdida de generalidad, tomamos un vértice no central a . Luego debe valer:

1. $d(a, f) = m < h = d(a, b)$, directamente por Lema 3.13.
2. $d(a, f) = m < h = d(a, c)$, directamente por Lema 3.13.
3. $d(a, f) = m < h = d(a, d)$, pues $\text{sen}(72^\circ) = \frac{d(a,d)/2}{m} \Leftrightarrow m = \frac{d(a,d)}{2 \cdot \text{sen}(72^\circ)}$ con $2 \cdot \text{sen}(72^\circ) > 1$.
4. $d(a, f) = m < h = d(a, e)$, por simetría con el punto (3).

Análogamente sucede parándose en b, c, d, e y revisando la desigualdad con las aristas $(b, f), (c, f), (d, f)$ y (e, f) .

Finalmente, para todos $v, w, z \in V$ tal que $(v, w) \in E$ y $(v, z) \notin E$ ocurre $d(v, w) < d(v, z)$. Entonces el grafo $K_{1,5}$ es *friend-enemy*. \square

Lema 3.16. *Los grafos split $K_{1,n}$ con $n \geq 6$ no tienen representación friend-enemy.*

Demostración. Supongamos que existiera un $K_{1,n}$ *friend-enemy* con $n \geq 6$, luego si v es el vértice central que se conecta con todos los v_1, \dots, v_n , por el Lema 3.7 debe ocurrir $\alpha_i = v_i \hat{v} v_{i+1} > 60^\circ$ para todo $1 \leq i < n$ y además $\alpha_n = v_n \hat{v} v_1 > 60^\circ$. La suma de los ángulos de la circunferencia desde v debe dar 360° , pero como $n \geq 6$, ocurre $\alpha_1 + \dots + \alpha_n > 360^\circ$, lo cual lleva a un absurdo. \square

Lema 3.17. *Sea un grafo split $G = (V, E)$ de clique máxima $K_2 = \{v, u\}$. Si v o u se conectan a más de 4 vértices del conjunto independiente $C = V \setminus \{v, u\}$ (por medio de aristas positivas), entonces G no tiene representación friend-enemy en \mathbb{R}^2 .*

Demostración. Supongamos que $gr(v) \geq 6$ (5 o más pertenecientes al conjunto C y 1 por su vecino u). Dado que todos los vecinos de v que están en $V \setminus \{v, u\}$ forman un conjunto independiente y no son adyacentes a u (pues de lo contrario se formaría un clique K_3), sucede que el subgrafo inducido por v y su vecindad directa resulta un $K_{1,gr(v)}$ con $gr(v) \geq 6$, lo cual vimos en el Lema 3.15 que no es *friend-enemy*. Entonces por el Lema 3.3, G no puede tener representación *friend-enemy*. \square

Lema 3.18. *Sea un grafo split $G = (V, E)$ de clique máxima $K_2 = \{v, u\}$ y en donde ambos vértices tienen 4 vecinos distintos (unidos por aristas positivas): $N(v) \setminus \{u\} = \{a_1, \dots, a_4\}$, $N(u) \setminus \{v\} = \{b_1, \dots, b_4\}$. Entonces G tiene representación friend-enemy.*

Demostración. Tomaremos la siguiente representación de G : todos los ángulos internos desde v a sus vecinos y desde u a sus vecinos deben ser mayores a 60° . Por otro lado, tomamos $d(v, u) = 3m$ y el resto de las aristas de largo m . Es claro que v y sus vecinos cumplen la propiedad *friend-enemy* (pues forman un $K_{1,5}$); de igual forma sucede entre

u y sus vecinos. En la Figura 3.4 se puede ver una representación a escala. Además, elegimos arbitrariamente $\alpha = 61^\circ$ y por lo tanto los ángulos restantes son de $88,5^\circ$. Nuestro objetivo será probar que esta representación es *friend-enemy*.

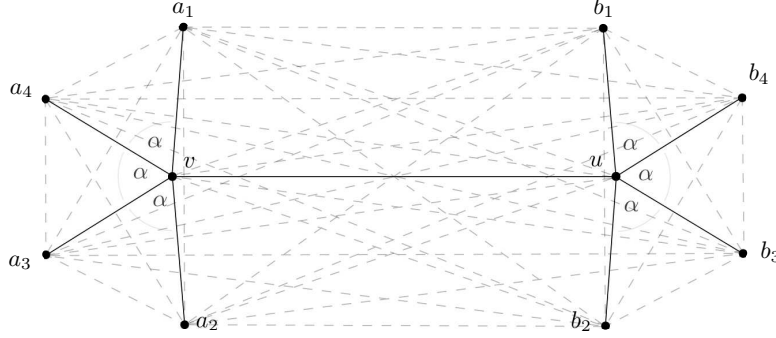


Fig. 3.4: Representación *friend-enemy* a escala sobre \mathbb{R}^2 del grafo del Lema 3.17

Analicemos que se cumplan las distancias entre vértices. En primer lugar, sabemos que entre los vértices de $\{v, a_1, a_2, a_3, a_4\}$ se cumple la propiedad pues los ángulos que los separan son de más de 60° . Lo mismo ocurre entre los vértices de $\{u, b_1, b_2, b_3, b_4\}$. Por otro lado, claramente se ve que las distancias $d(a_x, v)$ (con $x \in \{3, 4\}$) son menores que $d(a_x, b_y)$, $d(a_x, u)$, y $d(v, b_y)$ (con $y \in \{1, 2, 3, 4\}$); así como $d(a_1, v) < d(a_z, b_x)$ y $d(b_z, u) < d(b_z, a_x)$ (con $z \in \{1, 2\}$).

Dicho esto, nos vamos a enfocar en el esquema de la Figura 3.5a, en donde a continuación analizaremos las implicancias importantes tomando como punto de partida a la arista positiva (a_1, v) , como se muestra en la Figura 3.5b.

- $d(v, a_1) < d(v, b_1)$: la proyección de b_1 sobre la recta conformada entre v y u y medida desde u , tiene longitud menor a m (pues $d(u, b_1) = m$ y $\hat{v}ub_1 > 0^\circ$). Entonces $d(v, b_1) > 2m$ y luego $d(v, a_1) = m < 2m < d(v, b_1)$.
- $d(a_1, v) < d(a_1, b_1)$: con una idea análoga al punto anterior entre a_1 y b_1 , debe ocurrir $d(a_1, b_1) > m = d(a_1, v)$.
- $d(v, a_1) < d(v, b_2)$: por ser una representación simétrica horizontal y verticalmente y por el primer ítem, ocurre que $d(v, b_2) = d(v, b_1) > 2m$. Luego $d(v, a_1) = m < 2m < d(v, b_2)$.
- $d(a_1, v) < d(a_1, b_2)$: consideremos a l en donde $2l = d(a_1, a_2)$ (es decir, la mitad de la línea punteada vertical entre a_1 y a_2 ; o dicho de otro modo el camino que hace a_1 hacia a_2 hasta toparse exactamente con la recta entre v y u). Por trigonometría, $\text{sen}(88,5^\circ) = \frac{l}{m} \Leftrightarrow l = m \cdot \text{sen}(88,5^\circ)$. Ahora, observemos el triángulo formado por a_1, b_1 y b_2 : como $\hat{a}_1b_1b_2 = 90^\circ$, ocurre que (a_1, b_2) es la hipotenusa y por ende debe ser mayor que el cateto (b_1, b_2) , que mide $2l$. Como $2l = 2m \cdot \text{sen}(88,5^\circ) > m$, luego $d(a_1, b_2) > 2l > m$ y finalmente debe ocurrir $d(a_1, b_2) > m = d(a_1, v)$.

Con estos casos corroboramos que se cumpla la propiedad sobre las distancias de la

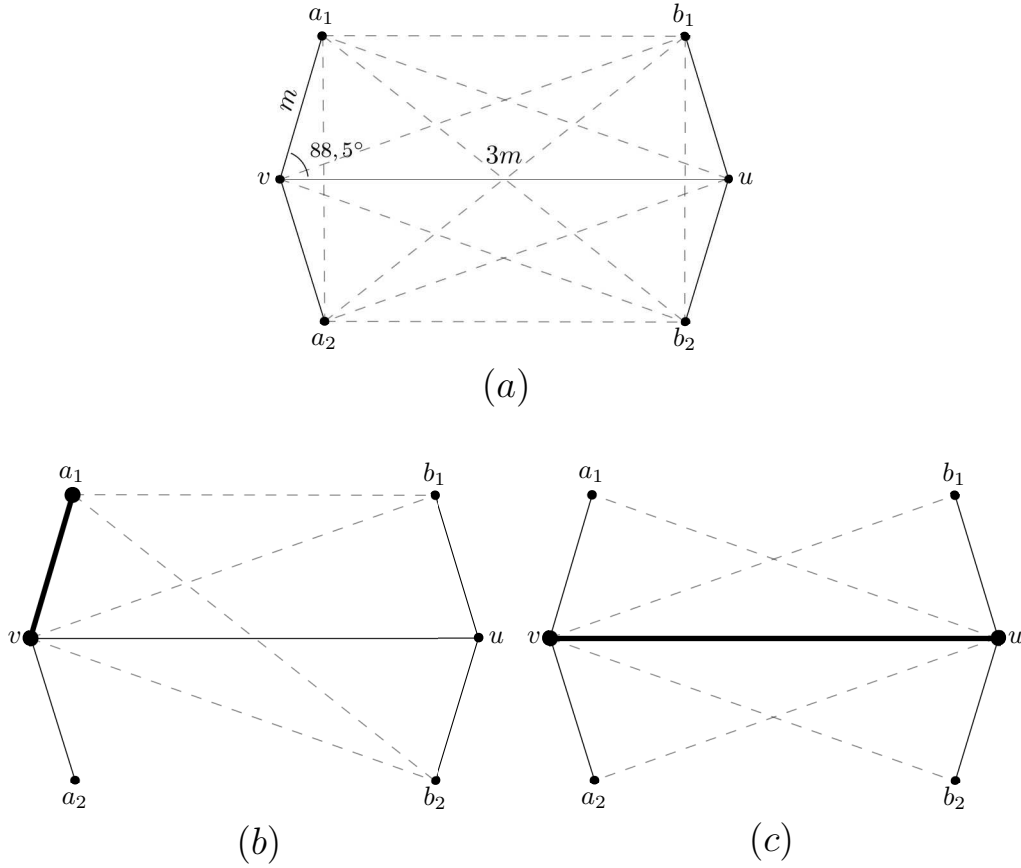


Fig. 3.5: Representación *no a escala* de distintas vistas del grafo de la Figura 3.4.

Figura 3.5b. Vemos que son mayores que $d(v, a_1)$, por lo cual es trivial que $d(v, a_1) < d(v, b_y)$ y $d(a_1, v) < d(a_1, b_y)$ (para $y \in \{1, 2, 3, 4\}$).

Además, vale esto mismo con $d(v, a_2) < d(v, b_y)$ y $d(a_1, v) < d(a_2, b_y)$ (con $y \in \{1, 2, 3, 4\}$) por simetría de la representación. Lo mismo parándose en u, b_1 y b_2 con respecto a v, a_1 y a_2 : $d(u, b_x) < d(u, a_y)$ y $d(b_x, u) < d(b_x, a_y)$ (para $x \in \{1, 2\}$ y $y \in \{1, 2, 3, 4\}$).

Resta ver ubicándose desde la arista (v, u) , como se ve en la Figura 3.5c. Dado que la representación es simétrica, ocurre que $a_1 \hat{v}u = a_2 \hat{v}u = b_1 \hat{u}v = b_2 \hat{u}v = 88,5^\circ$ y por eso sólo probaremos el caso $d(u, v) < d(u, a_1)$. Los demás casos son exactamente iguales.

Aplicando trigonometría sobre el triángulo la Figura 3.6, se puede ver que $d(u, a_1) > 3m$, por lo siguiente.

$$\begin{aligned}
 d(u, a_1)^2 &= (m \cdot \text{sen}(88,5^\circ))^2 + (3m - m \cdot \text{cos}(88,5^\circ))^2 \\
 d(u, a_1) &= \sqrt{m^2 \cdot [\text{sen}^2(88,5^\circ) + (3 - \text{cos}(88,5^\circ))^2]} \\
 &= 3m \cdot \sqrt{1 + \frac{\text{sen}^2(88,5^\circ) + \text{cos}^2(88,5^\circ) - 6\text{cos}(88,5^\circ)}{9}}
 \end{aligned} \tag{3.1}$$

Y como el interior de la raíz cuadrada tiene un valor mayor que 1, vale $d(u, a_1) > 3m$ y finalmente $d(u, v) = 3m < d(u, a_1)$.

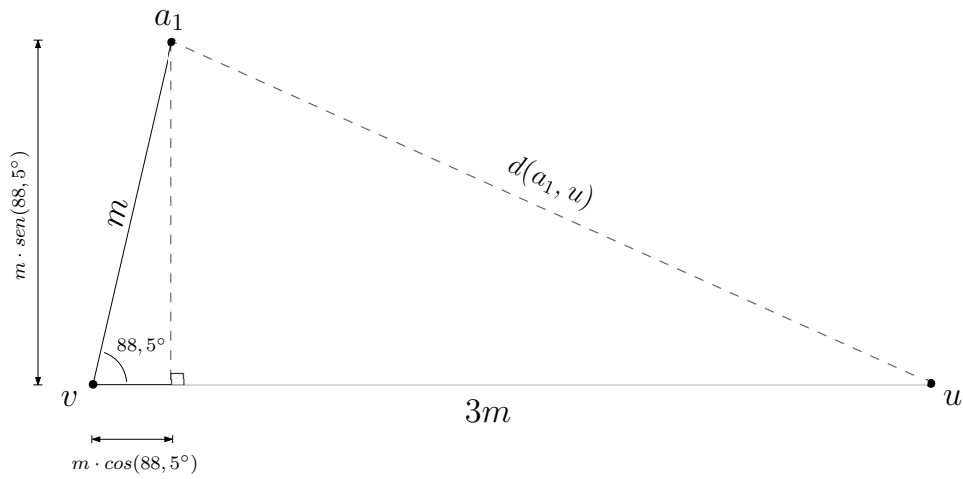


Fig. 3.6: Representación *no a escala* de relaciones de amistad y enemistad entre vértices a_1, v y u del grafo del Lema 3.17.

Dado que éstas son todas las posibilidades, el grafo en cuestión tiene representación *friend-enemy*. □

Como síntesis de las propiedades presentadas en esta sección, el Lema 3.17 presenta el grafo de la Figura 3.4. El mismo engloba todas las propiedades anteriores y determina todos los *split* signados completos con clique máximo K_2 que son *friend-enemy* sobre \mathbb{R}^2 (él mismo y todos sus posibles subgrafos inducidos), así como los que no, siendo $K_{1,6}$ el único subgrafo inducido prohibido en este caso.

3.3.2. Ideas sobre grafos *split* con clique máximo K_n , con $n \geq 3$

En esta última sección del capítulo presentaremos algunas propiedades generales que deben cumplir los *split* signados completos con representación *friend-enemy* que tengan como mínimo un K_3 como clique máximo, formado por aristas positivas.

Lema 3.19. *Sea una representación en el plano con dos vértices adyacentes v, u que se unen por aristas positivas a tres vértices vecinos en común provenientes de un conjunto independiente $C = \{c_1, c_2, c_3\}$, los cuales se unen entre sí por aristas negativas. Entonces si c_3 se encuentra en el área conformada por el polígono vc_1uc_2 , no resulta una representación *friend-enemy*.*

Demostración. Podemos representar esta situación como se muestra en la Figura 3.7.

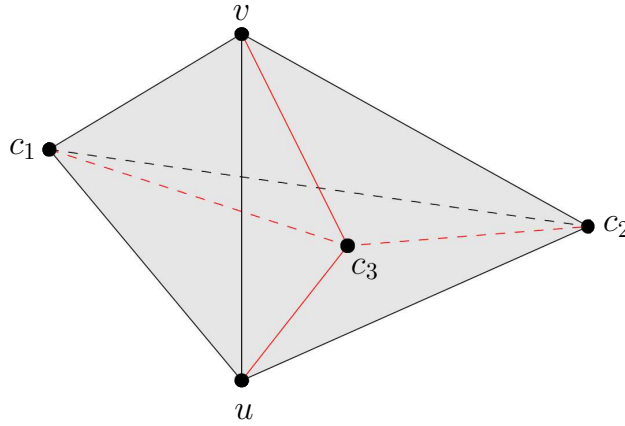


Fig. 3.7: Representación en donde se ubicaron v, u, c_1 y c_2 y se intenta que c_3 quede ubicado dentro del área.

Supongamos que existiera una representación *friend-enemy* que cumpla con la hipótesis y asumamos que están ubicados v, u, c_1 y c_2 conformando un polígono y que c_3 está en algún lugar dentro del área, por lo cual puede quedar ubicado sobre el triángulo vc_1u o sobre el vc_2u .

Sea el triángulo vc_iu (para $1 \leq i \leq 2$), luego para que todo resulte una representación *friend-enemy* mínimamente se debe cumplir que para cualquier $w \in C \setminus \{c_i\}$, suceda $d(c_i, v) < d(c_i, w)$ y $d(c_i, u) < d(c_i, w)$, es decir, que $m = \max\{d(c_i, v), d(c_i, u)\} < d(c_i, w)$, lo que es lo mismo que pedir que w no esté dentro del círculo de centro c_i y radio m . Pero es claro que esta circunferencia abarca completamente el triángulo vc_iu . Como supusimos que c_3 cumple la propiedad para con w , luego c_3 no puede estar dentro del triángulo vc_iu ; pero entonces no puede estar dentro del polígono vc_1uc_2 , lo cual lleva al absurdo. \square

Lema 3.20. *Un grafo *split* completo con partición $\{V_1, V_2\}$ tal que $|V_1| = 2$ es *friend-enemy* en \mathbb{R}^2 si y solo si $|V_2| \leq 5$.*

Demostración. Sea $V_1 = \{v, u\}$ y comencemos probando que dicho grafo no puede ser *friend-enemy* si $|V_2| > 5$. Si v y u tuvieran 6 o más vecinos en común del conjunto independiente V_2 , el subgrafo inducido dado por $V_2 \cup \{v\}$ formaría un $K_{1,n}$ con $n \geq 6$, lo cual por el Lema 3.15 no tiene representación *friend-enemy* en el plano y por lo tanto agregar

a u no cambiaría esto. Lo mismo sucede tomando el subgrafo dado por $V_2 \cup \{u\}$ y luego agregando v . Queda demostrado entonces que con 6 o más vecinos no existe representación *friend-enemy* en \mathbb{R}^2 .

Veamos entonces que para $|V_2| \leq 5$ se trata de un grafo *friend-enemy* en \mathbb{R}^2 . La idea será demostrar que funciona para un conjunto independiente de 5 elementos, $V_2 = \{c_1, c_2, c_3, c_4, c_5\}$ y luego derivar que con 4 o menos debe funcionar. Para esto utilizaremos el grafo de la Figura 3.8a, del cual puede verse una representación *friend-enemy* a escala en la Figura 3.8b.

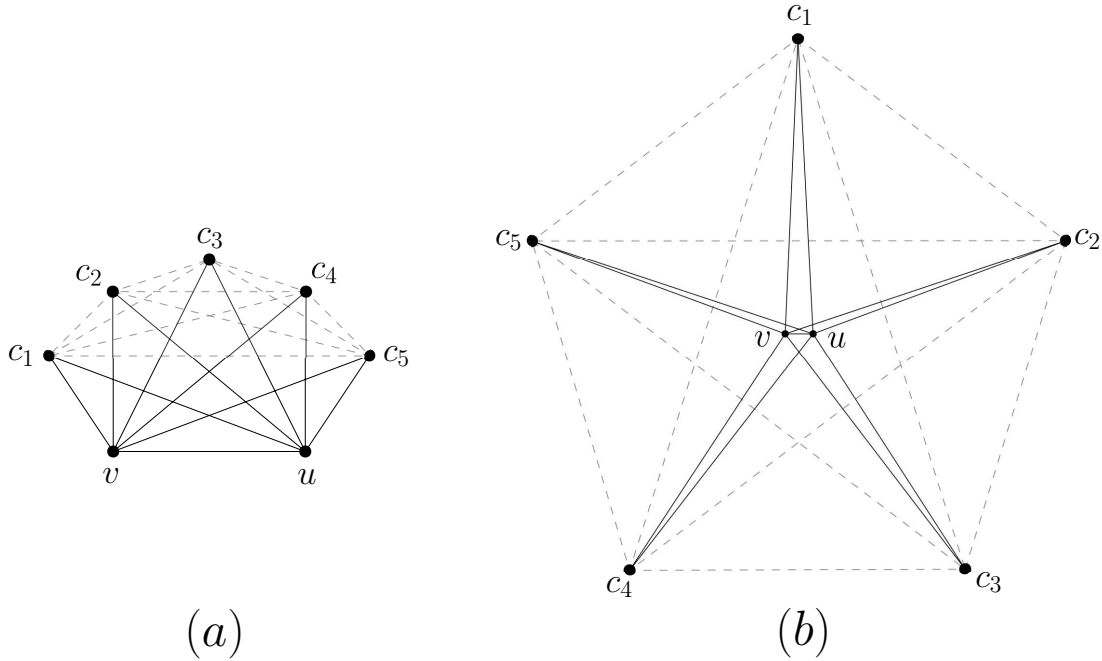


Fig. 3.8: Representación del grafo base en (a) y una representación *friend-enemy* del mismo en (b).

La representación de la Figura 3.8b puede obtenerse partiendo del grafo ya visto en la Figura 3.3a, es decir, un $K_{1,5}$ con forma de pentágono regular con ángulos $\alpha = 72^\circ$ y aristas positivas de longitud m . Luego, debe reemplazarse el vértice central por el centro de la arista (v, u) , en donde v y u quedarán ubicados a $\frac{m}{10}$ a la izquierda y a la derecha del centro, respectivamente. Una vez ubicados, quedarán dadas todas las conexiones entre $\{v, u\}$ y V_2 .

Veamos a continuación que dicho dibujo es una representación *friend-enemy*.

- Nos ubicamos en la perspectiva de c_1 . Sucede que, por la simetría de la representación, $d(c_1, v) = d(c_1, u)$. Tomamos en cuenta al triángulo rectángulo dado por c_1, u y el punto central del grafo (o lo que es lo mismo, el punto medio exacto de la arista entre v y u). Usando que el cateto vertical tiene largo m y un poco de trigonometría, $d(c_1, u)^2 = m^2 + \left(\frac{m}{20}\right)^2 = m^2 \cdot \frac{401}{400}$, por lo que $d(c_1, v) = d(c_1, u) = m \cdot \frac{\sqrt{401}}{20}$. Sucede que, dados c_i, c_{i+1} consecutivos, $d(c_i, c_{i+1}) = 2m \cdot \text{sen}(36^\circ)$. Entonces $d(c_1, v) = d(c_1, u) = m \cdot \frac{\sqrt{401}}{20} < 2m \cdot \text{sen}(36^\circ) < d(c_1, c_2)$, que por simetría también aplica que $d(c_1, v) = d(c_1, u) < d(c_1, c_5)$. A su vez, $d(c_1, c_3) = 2m \cdot \text{sen}(72^\circ)$,

por lo cual $d(c_1, v) = d(c_1, u) = m \cdot \frac{\sqrt{401}}{20} < d(c_1, c_3)$, que por simetría también aplica $d(c_1, v) = d(c_1, u) < d(c_1, c_4)$.

- Nos ubicamos en la perspectiva de c_2 . Llamemos w al punto central del grafo y consideremos a t_1 como el triángulo de vértices c_2, u y w , y a t_2 como el triángulo de vértices c_2, v, w . Aplicando trigonometría, obtenemos que $v\hat{w}c_2 = 162^\circ$ y $u\hat{w}c_2 = 18^\circ$. Usando la Ley de los Cosenos sobre t_2 , obtenemos

$$\begin{aligned} d(c_2, v)^2 &= \left(\frac{m}{20}\right)^2 + m^2 - 2\frac{m^2}{20} \cdot \cos(162^\circ) \\ &= m^2 \cdot \left(\frac{401}{400} - \frac{1}{10} \cdot \cos(162^\circ)\right) \\ d(c_2, v) &= m \cdot \frac{\sqrt{401 - 40\cos(162^\circ)}}{20} \end{aligned} \quad (3.2)$$

De la misma forma, aplicando la Ley de los Cosenos sobre t_1 , obtenemos

$$d(c_2, u) = m \cdot \frac{\sqrt{401 - 40\cos(18^\circ)}}{20} \quad (3.3)$$

por lo que, revisando los valores finales, obtenemos que $d(c_2, u) < d(c_2, v) < d(c_2, c_1) = d(c_2, c_3) < d(c_2, c_4) = d(c_2, c_5)$.

- Nos ubicamos en la perspectiva de c_3 . Con una idea análoga a la del punto anterior, tomamos los triángulos t_1 comprendido por los vértices c_3, u y w , y t_2 comprendido por c_3, v, w . Aplicando trigonometría, obtenemos que $v\hat{w}c_3 = 126^\circ$ y $u\hat{w}c_3 = 54^\circ$. Usando la Ley de los Cosenos sobre t_2 y t_1 , obtenemos respectivamente

$$d(c_3, v) = m \cdot \frac{\sqrt{401 - 40\cos(126^\circ)}}{20} \quad (3.4)$$

$$d(c_3, u) = m \cdot \frac{\sqrt{401 - 40\cos(54^\circ)}}{20} \quad (3.5)$$

por lo que, revisando los valores finales, obtenemos que $d(c_3, u) < d(c_3, v) < d(c_3, c_2) = d(c_3, c_4) < d(c_3, c_1) = d(c_3, c_5)$.

- Ubicados desde c_4 resulta idéntico al caso de c_3 invertido, pues

$$d(c_4, v) = m \cdot \frac{\sqrt{401 - 40\cos(54^\circ)}}{20} \quad (3.6)$$

$$d(c_4, u) = m \cdot \frac{\sqrt{401 - 40\cos(126^\circ)}}{20} \quad (3.7)$$

por lo que, obtenemos que $d(c_4, v) < d(c_4, u) < d(c_4, c_3) = d(c_4, c_5) < d(c_4, c_1) = d(c_4, c_2)$.

◦ Finalmente, ubicados desde c_5 resulta idéntico al caso de c_2 invertido, pues

$$d(c_5, v) = m \cdot \frac{\sqrt{401 - 40\cos(18^\circ)}}{20} \quad (3.8)$$

$$d(c_5, u) = m \cdot \frac{\sqrt{401 - 40\cos(162^\circ)}}{20} \quad (3.9)$$

por lo que, obtenemos que $d(c_5, v) < d(c_5, u) < d(c_5, c_1) = d(c_5, c_4) < d(c_5, c_2) = d(c_5, c_3)$.

Dado que todos los puntos críticos de esta representación cumplen correctamente (y las demás combinaciones ya fueron demostradas los Lemas 3.13 y 3.14), queda demostrado que esta representación es *friend-enemy*. □

Una vez presentada la propiedad anterior, usaremos dicha representación en el siguiente teorema.

Teorema 3.21. *Un grafo split completo G con partición $\{V_1, V_2\}$ es friend-enemy en \mathbb{R}^2 si y solo si $|V_2| \leq 5$.*

Demostración. Utilizaremos la representación de la Figura 3.8b para mostrar la idea de la demostración. Usaremos inducción en la cantidad de vértices p del clique V_1 , $p \in \mathbb{N}$.

Caso base: Si $p = 1$, el grafo en su totalidad resulta un $K_{1,m}$ con $m \leq 5$, por lo cual es *friend-enemy* por el Lema 3.14.

Si $p = 2$ y suponiendo el caso límite $|V_2| = 5$, entonces por el Lema 3.19 resulta *friend-enemy*.

Hipótesis inductiva: Dado el grafo *split completo* G de partición $\{V_1, V_2\}$ (con $|V_1| = p$ y $|V_2| \leq 5$), entonces G tiene representación *friend-enemy* como la usada en el Lema 3.19, agregando los $p - 2$ vértices restantes de V_1 entre medio del segmento \overline{vu} (sin repetir coordenadas de vértices).

Paso inductivo: Supongamos el caso límite $|V_2| = 5$ y sea w un vértice cualquiera de V_1 (con $|V_1| = p + 1$). Llamemos v y u a los vértices extremos (izquierda y derecha respectivamente) de la recta formada por los vértices de $V_1 \setminus \{w\}$. Por la hipótesis inductiva, el subgrafo inducido $V_1 \cup V_2 \setminus \{w\}$ tiene una representación *friend-enemy* como la usada en el Lema 3.19, con $p - 2$ vértices sobre el segmento \overline{vu} y sin repetir coordenadas. Luego se agrega a w en una coordenada libre dentro de \overline{vu} (debe existir pues hay infinitas posiciones), con w estrictamente entre v y u . Entonces se puede ver trivialmente que:

- Desde la perspectiva de c_1 , sabemos por lo visto en el Lema 3.19 que $d(c_1, v) = d(c_1, u) < d(c_1, c_2) = d(c_1, c_5) < d(c_1, c_3) = d(c_1, c_4)$. Además, como w está contenido en \overline{vu} , sucede que $d(c_1, w) < d(c_1, v) = d(c_1, u)$.
- Desde la perspectiva de c_2 , sabemos por el Lema 3.19 que $d(c_2, u) < d(c_2, v) < d(c_2, c_1) = d(c_2, c_3) < d(c_1, c_4) = d(c_1, c_5)$. Además, es fácil ver que $d(c_2, u) < d(c_2, w) < d(c_2, v)$.

- Desde la perspectiva de c_3 , sabemos por el Lema 3.19 que $d(c_3, u) < d(c_3, v) < d(c_3, c_2) = d(c_3, c_4) < d(c_3, c_1) = d(c_3, c_5)$. Además, es fácil ver que $d(c_3, u) < d(c_3, w) < d(c_3, v)$.
- De forma análoga e invertida respecto de c_3 , desde la perspectiva de c_4 se cumple $d(c_4, v) < d(c_4, w) < d(c_4, u) < d(c_4, c_3) = d(c_4, c_5) < d(c_4, c_1) = d(c_4, c_2)$.
- De forma análoga e invertida respecto de c_2 , desde la perspectiva de c_5 se cumple $d(c_5, v) < d(c_5, w) < d(c_5, u) < d(c_5, c_1) = d(c_5, c_4) < d(c_5, c_2) = d(c_5, c_3)$.

Por lo cual, dado que el dibujo del grafo sin w era una representación *friend-enemy*, agregar a w sólo requiere verificar las distancias en las que influye w , que son todas las anteriores. Por lo tanto el grafo *split completo* de partición $\{V_1, V_2\}$ con $|V_2| = 5$ es *friend-enemy* sobre el plano.

Finalmente, por el Lema 3.3 vale que si $|V_2| \leq 5$, el grafo *split completo* de partición $\{V_1, V_2\}$ es *friend-enemy* sobre \mathbb{R}^2 para todo $p = |V_1|$.

□

4. CONCLUSIONES Y TRABAJO FUTURO

A lo largo de este trabajo estudiamos el estado actual de investigación de los grafos que pueden ser dibujados sobre la recta y el plano, utilizando la distancia Euclídea, de forma que para todo vértice, los vecinos queden ubicados más cerca que cualquier no-vecino. A estos grafos los denominamos *friend-enemy*.

Demostremos el caso de algunos grafos presentados en [14] que no son *friend-enemy* sobre \mathbb{R} , como el *positive square*, el *positive star*, el *positive triangle* y algunas familias infinitas.

Estudiamos los grafos signados completos junto con algunos algoritmos que permiten determinar si uno de estos grafos tiene representación sobre la recta. El problema en este caso resulta polinomial y sencillo de resolver pues resulta equivalente a encontrar una secuencia de eliminación perfecta y testear que dicha secuencia cumpla con una propiedad combinatoria adicional; por lo cual atacamos el problema en primer lugar por el lado de los grafos *cordales*. Luego exhibimos una alternativa teórica propuesta en [4] para atacar el problema mediante grafos de *intervalos propio*.

Luego analizamos qué sucede cuando se quiere representar sobre la recta a un grafo signado incompleto (es decir, que soporta la indiferencia entre vértices) y encontramos que el problema se vuelve NP-Completo. Presentamos la equivalencia de dicho problema, que denominamos LINE CLUSTER EMBEDDING, con otros ya conocidos como ACYCLIC PARTITION, SET SPLITTING y finalmente 3-CNF-SAT. Estudiamos una cota de complejidad temporal mínima para la resolución del problema mediante un algoritmo exponencial, asumiendo válido el *Exponential Time Hypothesis* y por último estudiamos un algoritmo exponencial que resuelve el caso general de dibujar a un grafo signado sobre la recta.

Una vez terminado el problema sobre la recta, analizamos qué sucede con la propiedad sobre \mathbb{R}^2 al usar grafos signados completos. Presentamos ejemplos de grafos que no tienen representación, como el *negative triangle* y de grafos que sí la tienen como el positivo $K_{1,5}$. Aportamos varias propiedades generales, como que la propiedad *friend-enemy* es hereditaria (se mantiene en subgrafos inducidos) o que dados dos vértices enemigos unidos por un amigo en común tienen que conformar un ángulo mayor a 60° .

Finalmente abordamos este último problema sobre la clase de grafos *split*, en la cual pudimos caracterizar por medio de subgrafos prohibidos a todos los *split* con clique máximo positivo K_2 y aportamos algunas propiedades para los *split* con clique K_n y $n \geq 3$, como ser que cualquier grafo *split completo* es *friend-enemy* en \mathbb{R}^2 si y solo si el conjunto independiente tiene a lo sumo 5 vértices.

El problema de caracterizar a los grafos *friend-enemy* es un terreno casi inexplorado sobre espacios \mathbb{R}^n con $n \geq 2$, por lo cual surgen varias preguntas que uno puede hacerse, como ser en principio *¿existe un algoritmo polinomial para determinar si un grafo signado completo se puede representar bajo esta propiedad sobre \mathbb{R}^2 ?, ¿se puede caracterizar de una forma eficiente teóricamente (de forma mínima, por ejemplo) a los grafos que poseen esta propiedad sobre \mathbb{R}^2 ?, ¿qué familias de grafos vuelven el problema en cuestión polinomial para grafos signados completos?... ¿y para grafos signados incompletos?*

Bibliografía

- [1] ANTAL, T., KRAPIVSKY, P. L., AND REDNER, S. Dynamics of social balance on networks. *Phys. Rev. E* 72(3) (2005).
- [2] BRANDES, U., FLEISCHER, D., AND LERNER, J. Summarizing dynamic bipolar conflict structures. *IEEE Trans. Vis. Comput. Graph* 12(6) (2006), 1486–1499.
- [3] CARTWRIGHT, D., AND HARARY, F. Structural balance: a generalization of heider’s theory. *Psychological Review* 63(5) (1956), 277–293.
- [4] CYGAN, M., PILIPCZUK, M., PILIPCZUK, M., AND WOJTASZCZYK, J. O. Sitting closer to friends than enemies, revisited. *Theory of Computing Systems* 56 (2015), 394–405.
- [5] EPPSTEIN, D., AND MUMFORD, E. Self-overlapping curves revisited. *SODA ’09 Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms* (2009), 160–169.
- [6] GAREY, R., AND JOHNSON, D. Computers and Intractability: A guide to the theory of NP-Completeness. *W. H. Freeman and Company* (1979).
- [7] GOLUMBIC, M. C. Algorithmic Graph Theory and Perfect Graphs, 2 ed. *Annals of Discrete Mathematics* 57 (2004).
- [8] GUILLEMOT, S., JANSSON, J., AND SUNG, W.-K. Computing a smallest multilabeled phylogenetic tree from rooted triplets. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 8(4) (2011), 1141–1147.
- [9] HABIB, M., MCCONNELL, R. M., PAUL, C., AND VIENNOT, L. Lex-bfs and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing. *Theor. Comput. Sci* 234 (1-2) (2000), 59–84.
- [10] HEIDER, F. Attitudes and cognitive organization. *J. Psychol.* 21 (1946), 107–112.
- [11] IMPAGLIAZZO, R., AND PATURI, R. On the complexity of k-sat. *J. Comput. Syst. Sci.* 62(2) (2001), 367–375.
- [12] IMPAGLIAZZO, R., PATURI, R., AND ZANE, F. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.* 63(4) (2001), 512–530.
- [13] KAHN, A. B. Topological sorting of large networks. *Communications of the ACM* 5(11) (1962), 558–562.
- [14] KERMARREC, A. M., AND THRAVES, C. Can everybody sit closer to their friends than their enemies? *Lecture Notes in Computer Science* 6907 (2011), 388–399.
- [15] KERMARREC, A. M., AND THRAVES, C. Signed graph embedding: when everybody can sit closer to friends than enemies. *arXiv:1405.5023v1* (2015).

-
- [16] LAUTERBACH, D., TRUONG, H., SHAH, T., AND ADAMIC, L. A. Surfing a web of trust: Reputation and reciprocity on couchsurfing.com. *CSE 4* (2009), 346–353.
- [17] LESKOVEC, J., HUTTENLOCHER, D. P., AND KLEINBERG, J. M. Governance in social media: A case study of the wikipedia promotion process. *ICWSM 2010* (2010).
- [18] LESKOVEC, J., HUTTENLOCHER, D. P., AND KLEINBERG, J. M. Predicting positive and negative links in online social networks. *WWW 2010* (2010), 641–650.
- [19] LESKOVEC, J., HUTTENLOCHER, D. P., AND KLEINBERG, J. M. Signed networks in social media. *CHI 2010* (2010), 1361–1370.
- [20] LOOGES, P. J., AND OLARIU, S. Optimal greedy algorithms for indifference graphs. *Comput. Math. Appl.* 25(7) (1993), 15–25.
- [21] OLARIU, S. An optimal greedy heuristic to color interval graphs. *Information Processing Letter* 37 (1991), 21–25.
- [22] PARDO, E. G., SOTO, M., AND THRAVES, C. Embedding signed graphs in the line. *Journal of Combinatorial Optimization* (2013), 1–21.
- [23] ROBERTS, F. S. Graph theory and its application to problems of society. *SIAM Press* (1978).
- [24] ROSE, D. J., TARJAN, R. E., AND LEUKER, G. S. Algorithmic aspects of vertex elimination on graphs. *SIAM J. Comput.* 5 (1976), 266–283.
- [25] SCHMITT, B. R. Triple alliance and triple entente. *Holt, Rinehart, and Winston, New York* (1934).
- [26] SZELL, M., LAMBIOTTE, R., AND THURNER, S. Multirelational organization of large-scale social networks in an online world. *PNAS* 107(31) (2010), 13636–13641.