Problemas de dominación de aristas: algoritmos, cotas y propiedades

Moyano, Verónica Andrea

2017-03-29

UNIVERSIDAD DE BUENOS AIRES

Facultad de Ciencias Exactas y Naturales

Departamento de Matemática

Problemas de dominación de aristas: algoritmos, cotas y propiedades.

Tesis presentada para optar al título de Doctor de la Universidad de Buenos Aires en el área Ciencias Matemáticas

Verónica Andrea Moyano

Director de tesis: Min Chih Lin

Consejero de estudios: Guillermo Durán

Lugar de trabajo: Instituto de Cálculo, FCEyN, Universidad de Buenos Aires.

Buenos Aires, 2017

Agradecimientos

Quiero agradecer a mi director Oscar por confiar en mí, por todo el empuje y la ayuda en el doctorado, porque aprendí muchisimas cosas y porque es un excelente director y persona.

A Flavia, Valeria y Luerbio, por revisar esta tesis y por sus valiosos aportes y sugerencias.

A Willy por ser mi consejero de estudios. Al Instituto de Cálculo, el Departamento de Matemática, la FCEyN y sus profesores a los que debo mi formación académica. También gracias a Débora por ayudarme en todos mis trámites.

A mis compañeros (y amigos) del Instituto, porque realmente hacen que trabajar ahí sea un placer. Gracias por los almuerzos, los mates, las charlas y los consejos. Gracias Manu B., Pablo T., Chebi, Flor S., Marina, Maru, Andrés, Agus, Flor F.S., Nina, Michel, Saveli, Xavi, Manu S., Ines, Stella, Ale, Manu F., Lu, Ale, Maxi, Pablo V. y Lucas.

A mis amigos por bancarme hablando de grafos cada tanto (todo en esta vida se puede explicar con grafos) y por compartir la vida conmigo. Gracias Maggie, Pablo B., Juli C., Neru, Fede F., Marcos C., Maga K., Lucio, Pau, Mati S., Gara, Emilce, Natu, Rochis, Elina, y seguramente me estoy olvidando de algunos. A todos gracias, los quiero fuerte!

A mi familia, en especial a mi hermano Rodrigo, por todo su apoyo.

Finalmente, y no menos importante, gracias a la Agencia Nacional de Promoción Científica y Tecnológica y al CONICET por las becas que me permitieron dedicarme, casi exclusivamente, a este trabajo de investigación.

Edge dominating set problems: Algorithms, bounds and properties.

Abstract

In this thesis we study different dominating set problems with two different approaches: combinatoric and algorithmic. The first one consists in understanding the structure of the graph with respect to the minimum solution, and in counting the number of minimal solutions that the graph can contain. The algorithmic approach classifies the domination problem on different graph classes according to its complexity (NP-complete or polynomial-time solvable), while it also tries to develop efficient algorithms for the problems. The variants of domination problems we consider in this work are (i) edge domination, (ii) efficient edge domination, (iii) perfect edge domination, and (iv) vertex domination. Efficient edge dominating sets are also known in the literature as dominating induced matchings.

For (i) we present a linear time algorithm to find a minimum edge dominating set on proper interval graphs. For (ii) we prove tight bounds for the number of dominating induced matching, while we described the extremal graphs for the classes of general, triangle-free, and connected graphs. For (iii) we present a linear time algorithm to solve the weighted perfect edge dominating set problem for cubic claw-free graphs, and a robust linear time algorithm for P_5-free graphs.

Problem (i) is equivalent to (iv) restricted to line graphs, which form a subclass of $K_{1,3}$-free. We prove that (iv) is NP-Hard for well-covered $K_{1,4}$-free graphs, while it requires linear time for well-covered $K_{1,3}$-free graphs, which is a superclass of well-covered line graphs. Finally, we present polynomial time algorithms to decide if a comparability graph or its complement is well-covered.

Problemas de dominación de aristas: Algoritmos, cotas y propiedades.

Resumen

En esta tesis estudiamos problemas de conjunto dominante mediante dos enfoques diferentes: combinatorio y algorítmico. El primero consiste en entender las esctructuras del grafo relacionadas con la solución mínima y también contar el número de soluciones minimales que un grafo puede admitir. El enfoque algorítmico busca clasificar los problemas de dominación para diferentes clases de grafos de acuerdo a su complejidad (NP-completo o polinomial), mientras que también intenta desarrollar algoritmos eficientes que resuelvan estos problemas. Las variantes de problemas de conjunto dominante que consideramos en este trabajo son (i) dominación de aristas (ii) dominación eficiente de aristas (iii) dominación perfecta de aristas y (iv) dominación de vértices. En la literatura también se conoce a los conjuntos eficientemente dominantes con el nombre de matching inducidos dominantes.

Para el problema (i) presentamos un algoritmo de tiempo lineal para encontrar un conjunto dominante de aristas mínimo para los grafos de intervalos propios. Para el problema (ii) probamos cotas ajustadas para el número de matching inducidos dominantes y también describimos los grafos maximales para la clase general de grafos, grafos sin triángulos y grafos conexos. Para el problema (iii) presentamos un algoritmo de tiempo lineal para resolver el problema de dominación perfecta de aristas con pesos para los grafos cúbicos que no contienen garras, y un algoritmo robusto, también de tiempo lineal, para los grafos que no contienen P_5.

El problema (i) es equivalente a (iv) cuando nos restringimos a los grafos de líneas, estos grafos forman una subclase de los grafos que no contienen $K_{1,3}$. En la tesis probamos que el problema (iv) es NP-Hard para los grafos bien cubiertos que no contienen $K_{1,4}$, mientras el problema se resuelve en tiempo lineal para los grafos bien cubiertos que no contienen $K_{1,3}$, la cual es una superclase de los grafos bien cubiertos de línea. Finalmente, presentamos algoritmos polinomiales para decidir si un grafo de comparabilidad o su complemento es bien cubierto.

Contents

Introducción

En la actualidad, la teoría de grafos es una disciplina dinámica tanto en la teoría como en las aplicaciones. Los problemas de cubrimiento forman un campo en continuo crecimiento dentro de la teoría de grafos. Muchas de los problemas surgen de las aplicaciones y otros por intereses teóricos y de las conjeturas científicas. Los problemas de conjunto dominante, en particular dominantes de aristas, son los principales objetos de estudio en esta tesis.

Los grafos son excelentes herramientas para modelar problemas que pueden venir de distintos contextos pues sirven como representación de las relaciones entre objetos. Por ejemplo, una red de ciudades las cuales son representadas con vértices y sus conexiones representadas con aristas, definen un grafo. En algunos contextos, también hay una función de peso definida en los vértices y/o aristas. En el ejemplo anterior, el peso de una conexión puede representar la distancia entre las ciudades conectadas. El famoso problema del viajante de comercio busca el recorrido más corto posible que visite todas las ciudades exactamente una vez. Los problemas de conjuntos dominantes en grafos son modelos naturales para los problemas de locación de resursos en investigación operativa. Estos problemas buscan la locación de una o más facilidades de manera que se optimice cierto objetivo que puede ser minimizar el costo de transporte, ofrecer servicio equitativo a los clientes u obtener la mayor participación en el mercado. El concepto de dominación también se aplica en la teoría de códigos como lo muestran Kalbfleisch, Stanton y Horton en [52] y también Cockayne y Hedetniemi en [26]. Si se define un grafo, siendo los vértices los vectores n-dimensionales con coordenadas elegidas en $\{1, ..., p\}$, $p > 1$, y dos vértices son adyacentes si difieren en una coordenada, entonces los conjuntos que son (n, p)-cubrimientos, códigos de corrección de errores simples, o los cubrimientos perfectos son todos conjuntos dominantes del grafo con propiedades adicionales.

Las conjeturas científicas son otra fuente de problemas interesantes y han tenido un impacto importante en el desarrollo de la teoría de grafos. Por ejemplo, Berge formuló la Conjetura Fuerte de los Grafos Perfectos en sus libros [3, 4] en los 60s. Por 40 años, los intentos por resolver esta conjetura dieron lugar a métodos poderosos, importantes conceptos y resultados interesantes en diferentes áreas de la teoría de grafos. Finalmente, la conjetura fue probada en 2006 [25] y ahora se conoce como el Teorema Fuerte de los Grafos Perfectos. En la teoría de dominación tenemos que mencionar la famosa y todavía abierta conjetura de Vizing formulada en 1963 [83]. La conjetura afirma que el tamaño de un conjunto dominante mínimo, llamado el *número de dominación*, del producto Cartesiano de dos grafos $F \square G$ es al menos el producto de los números de dominación de F y G. Las estrategias para resolver la conjetura incluyen familias de grafos para las cuales la conjetura es válida, propiedades de un supuesto contraejemplo minimal y conjeturas relacionadas. Un buen resumen del progreso en este campo se puede encontrar en [15].

Desde el punto de vista computacional, los problemas pueden ser clasificados de acuerdo a su dificultad siguiendo la teoría de la NP-completitud formalizada por Cook en [27]. Muchos de los problemas de cubrimiento en teoría de grafos pertenecen a la clase NP-completa. El hecho de que no sea posible resolver estos problemas en un tiempo razonable conduce a diferentes estrategias para dar soluciones que puedan servir a propósitos prácticos. Una de esas estrategias es restringir el dominio del problema. Debido a que muchas veces las entradas no son datos arbitrarios, se pueden hacer muchas suposiciones de acuerdo al campo de donde viene el problema y estas restricciones se formalizan con propiedades estructurales que toda entrada debe satisfacer. Muchas clases de grafos han sido definas siguiendo propósitos prácticos y muchas otras surgen de la teoría. Entre las primeras podemos mencionar los grafos de intersección [33] como los grafos de intervalos [11, 55], grafos arco circulares [11, 39], pero hay más. Otras clases de grafos son definidas con restricciones en parámetros como el máximo grado de sus vértices o número de independencia. Un resumen bien organizado sobre las familias de grafos estudiadas en la actualidad es [11]. En esta tesis pusismos especial esfuerzo en diseñar los algoritmos propuestos de manera eficiente y robusta para hacerlos útiles para propósitos prácticos.

Las preguntas teóricas sobre los problemas de cubrimiento pueden ser muy diferentes. Un enfoque es acotar el número de soluciones de un problema. Esto se relaciona con el punto de vista computacional pues da lugar a una cota natural para la búsqueda exhaustiva. Distintos parámetros relacionados y variantes de los problemas de conjunto dominante han sido definidos y ampliamente estudiados. También, este tipo de análisis es útil para entender las estructuras internas relacionadas y, a veces, desarrollar algoritmos eficientes para revolver los problemas.

Antecedentes

Las ideas teóricas de grafos se remontan al menos a la década de 1730, cuando Leonard Euler publica su trabajo sobre el problema de los siete puentes de Köningsberg [6]. En los comienzos gran parte de la teoría de grafos estuvo motivada por el estudio de juegos y la matemática recreativa. Los grafos son herramientas útiles para modelar relaciones entre objetos. La convención usual es que los objeros son representados por vértices (puntos) y las relaciones entre ellos son representadas por aristas (líneas) que unen esos vértices. En general, cualquier objeto matemático que involucre puntos y conexiones entre ellos puede ser llamado un grafo o hipergrafo. No hay restricción alguna sobre los objetos o sus relaciones, de esta manera los grafos pueden ser usados en bases de datos, redes físicas, flujo de señales, redes sociales o el flujo de un programa de computación, entre muchas otras aplicaciones. Normalmente, los problemas de la vida real pueden ser modelados con grafos que satisfacen varias propiedades, otras clases de grafos han sido definidas con propósitos teóricos. La ventaja de restringir el dominio de los problemas es que sea más fácil trabajar en una solución para esas representaciones restringidas. Un resumen acerca de las clases de grafos incluyendo la teoría de intersección se puede encontrar en [11].

Dado que el concepto de dominación en grafos surge naturalmente, hay referencias a problemas relacionados con dominación desde hace cientos de años. En 1862 de Jaenisch [29] y en 1892 W.W. Rouse Ball [78] presentaron diferentes problemas de dominación en tableros de ajedrez. La teoría de dominación fue desarrollada posteriormente en los 1950s y 1960s. En 1958 C. Berge escribió un libro sobre teoría de grafos [5] en el que introducía el *coeficiente de estabilidad externa*, que ahora es conocido como el número de dominación de un grafo. En el libro de O. Ore [70] los términos de *conjunto dominante* y *número de dominación* fueron introducidos, pero fue después de los avances de los hermanos Yaglom y Yaglom [86] y el trabajo de recopilación de Cockayne y Hedetniemi [26] que la notación $\gamma(G)$ fue usada por primera vez para el número de dominación de un grafo. Desde entonces, los conjuntos dominantes en grafos ha sido ampliamente estudiados. En [38] se demostró que el problema del conjunto dominante mínimo y muchas de sus interesantes variantes son problemas NP-completos. Resúmenes recientes sobre la teoría de dominación son [42, 46–48] y varias aplicaciones de problemas de conjuntos dominantes se pueden encontrar en [73].

En la siguiente sección, damos la terminología, definiciones y notaciones que usaremos a lo largo de esta tesis.

Notaciones y Definiciones

Un *grafo* $G = (V, E)$ consiste en un conjunto (finito), notado $V(G)$ o V, de *vértices* o *nodos*, y un conjunto $E(G)$ o E de pares de vértices, llamados *aristas* o *líneas*. Las aristas representan la relación entre los vértices. El número de vértices, $|V| = n$, es el *orden* del grafo, y $|E| = m$ es su *tamaño*. En el caso de *grafos no dirigidos*, las aristas son representadas por pares no ordenados de vértices, $e = uv \in E$. En los *grafos dirigidos*, cada arista (dirigida) es representada por un par ordenado de vértices $e = (u, v)$ o $e = \vec{uv}$. Cuando $e = uv \in E$, decimos que los vértices u y v son *adyacentes* o *vecinos*. Además, la arista e tiene extremos u y v, y e es *incidente* en u y en v. Dos aristas son *adyacentes* si son incidentes en un mismo vértice. Salvo que indiquemos lo contrario, todos los grafos que consideramos en esta tesis, son no dirigidos.

El conjunto de vecinos de un vértice $v \in V(G)$ se llama su *vecindario abierto* y se nota $N(v)$, mientras que $N[v] = N(v) \cup \{v\}$ es su *vecindario cerrado*. El *grado* de un vértice es $\deg(v) = |N(v)|$. El mínimo y máximo grado de los vértices en $V(G)$ son notados con $\delta(G)$ y $\Delta(G)$, respectivamente. Un grafo donde todos los vértices tienen el mismo grado r se llama *r-regular*. Los grafos 3-regulares también reciben el nombre de *cúbicos*. Para $S \subset V(G)$, $N(S) = \bigcup_{v \in S} N(v) \setminus S$, y $N[S] = \bigcup_{v \in S} N[v]$. Un vértice *universal* es un vértice que satisface $N[v] = V(G)$. Un conjunto de vértices no adyacentes dos a dos es un conjunto *independiente* o *estable*. Un *matching* es un conjunto de aristas no adyacentes.

Un *subgrafo inducido* de un grafo G es un grafo formado por un subconjunto de los vertices de G y todas las aristas de G con ambos extremos en ese subconjunto. Dado un subconjunto $S \subset V(G)$, notamos con $G[S]$ al subgrafo inducido por S. Un subgrafo H de G se dice *dominante* si los vértices de H, $V(H)$, es un conjunto dominante de G, es decir que todo vértice en $V(G) \setminus V(H)$ es adyacente a algún vértice en $V(H)$. Un grafo es *sin* H cuando H no es un subgrafo inducido de G. Un *conjunto completo* en un grafo es un subconjunto de vértices que inducen un grafo completo, esto es que todo par de vértices son adyacentes. Una *clique* es un conjunto completo maximal. El grafo completo de orden n se nota con K_n. El grafo K_3 también recibe el nombre de *triángulo*.

Un *camino* es una sucesión de aristas adyacentes que conectan una sucesión de vértices distintos entre sí. El número de aristas es el largo del camino. Un grafo es *conexo* cuando hay un camino que conecta u y v para cualquier par de vértices u y v; y una *componente conexa* es un subgrafo conexo maximal. Un *puente* es una arista e tal que si se borra, aumenta la cantidad de componentes conexas de G. La *distancia* entre dos vértices, notamos $dist(u, v)$, es el largo del camino más corto entre ellos. Análogamente, un *ciclo* es una sucesión de vértices adyacentes tal que sólo el primero y el último vértices son el mismo. El número de aristas es el largo del ciclo. Un camino (ciclo) es *inducido* cuando ninguna arista del grafo conecta dos vértices no consecutivos del camino (ciclo). Notamos con P_n y C_n al camino y al ciclo de n vertices, respectivamente.

Un grafo es *bipartito* cuando sus vértices se pueden particionar en dos conjuntos disjuntos e indepententes. Equivalentemente, G es bipartito cuando no contiene ciclos de longitud impar. Como siempre, K_{n_1,n_2} denota al grafo completo bipartito; esto es un grafo bipartito con vértices $V_1 \cup V_2$, $|V_1| = n_1$, $|V_2| = n_2$ donde todo vértice de V_1 es adyacente a todo vértice de V_2. Los grafos $K_{1,d}$ se llaman *estrellas*, y $K_{1,3}$ también recibe el nombre de *garra*.

Seguimos la notación de [66] para introducir el práctico concepto de grafos de intersección. Dada una familia de conjuntos $\mathcal{F} = \{S_1, ..., S_n\}$, el *grafo de intersección de* $\mathcal{F}$, $\Omega(\mathcal{F})$, es el grafo formado por los vértices $\mathcal{F}$ donde S_i y S_j son adyacentes si y sólo si tienen intersección no vacía $S_i \cap S_j \neq \emptyset$. Un grafo es un *grafo de intersección* si existe una familia $\mathcal{F}$ tal que G y $\Omega(\mathcal{F})$ son isomorfos; lo cual significa que hay una biyección entre los conjuntos de vértices que preserva la relación de adyacencia. En tal caso, $\mathcal{F}$ se llama una *representación* de G. Notar que un grafo puede admitir diferentes familias de representaciones.

Dado un grafo G, su *grafo de línea*, notado $L(G)$, es el grafo de intersección de las aristas de G.

El *cuadrado de un grafo* G, notado G^2, es el grafo con el mismo conjunto de vértices que G donde dos vértices son adyacentes en G^2 si y sólo si la distancia entre ellos en el grafo G es a lo sumo 2.

El *complemento de un grafo* $G = (V, E)$, es el grafo notado $\bar{G} = (V, \bar{E})$ con el mismo conjunto de vértices que G y donde dos vértices son adyacentes en $\bar{G}$ si y sólo si no son adyacentes en G.

Un *grafo de comparabilidad* $G = (V, E)$ es un grafo no dirigido tal que admite una orientación transitiva de sus aristas. Esto es, que cada arista recibe una orientación $\overrightarrow{uv}$ o $\overleftarrow{uv}$, y en el grado dirigido que queda definido, siempre que existan las aristas $\overrightarrow{uv}$ y $\overrightarrow{vw}$ entonces también existe la arista $\overrightarrow{uw}$.

Problemas de grafos

Dado un grafo G, $\alpha(G)$ denota el tamaño de un conjunto independiente máximo; es decir, con la mayor cantidad de vértices. Determinar $\alpha(G)$ es un problema NP-difícil [38].

Un grafo está *bien cubierto* si todos los conjuntos independientes maximales tienen el mismo tamaño. Probar si un grafo contiene dos conjuntos independientes maximales de diferente tamaño es un problema NP-completo; esto significa, complementariamente, que probar si un grafo está bien cubierto es coNP-completo [24, 79].

Un *cubrimiento de aristas por vértices* o *cubrimiento de aristas* de un grafo G es un subconjunto de vértices $S \subset V$ tal que toda arista incide en al menos un vértice de S. Equivalentemente, si $V \setminus S$ es independiente. Entonces, el problema de hallar un cubrimiento de aristas de mínimo tamaño es también NP-difícil.

Decimos que un vértice *domina* a sí mismo y a cualquier otro vértice adyacente a él.

Un *conjunto dominante de vértices* de un grafo $G = (V, E)$ es un subconjunto $D \subset V$ tal que todo vértice de G está dominado por algún vértice de D. Equivalentemente, todo vértice fuera de D es adyacente a al menos un vértice de D. El *número de dominación* $\gamma(G)$ es el número de vértices en un conjunto dominante mínimo. El problema del conjunto dominante consiste en deteminar si $\gamma(G) \leq k$ para un grafo G y k dados.

Análogamente, una arista *domina* a sí misma y a toda arista adyacente a ella. Un *conjunto dominante de aristas* de un grafo $G = (V, E)$ es un subconjunto $E' \subset E$ tal que toda arista de G está dominada por al menos una arista en E'. Equivalentemente, si toda arista fuera de E' es adyacente a al menos una arista en E'. El problema del conjunto dominante de aristas consiste en determinar la existencia de un conjunto dominante de aristas satisfaciendo $|E'| \leq k$ para un grafo G y k dados. Se probó que ambos problemas (dominación de vértices y de aristas) son NP-completos en [38] y [40], respectivamente. Salvo que indiquemos lo contrario 'dominación' siempre se refiere a la dominación de vértices.

A lo largo de esta tesis mencionamos algunas variantes de problemas de conjunto dominante. Definimos ahora esas variantes y damos más detalles y referencias más adelante.

- Un conjunto *perfectamente* dominante de un grafo $G = (V, E)$ es un subconjunto $D \subset V$ tal que todo vértice que no está en D es adyacente a exactamente un vértice de D. Equivalentemente, todo vértice en $V \setminus D$ está dominado por exactamente un vértice en D.
- Un conjunto *eficientemente* dominante, o conjunto *dominante independiente*, es un conjunto perfectamente dominante D tal que D es también un conjunto independiente. Equivalentemente, todo vértice está dominado por exactamente un vértice en D. El *número de dominación independiente*, $i(G)$, es el número de vértices en un conjunto dominante indpendiente mínimo de G.

Análogamente, para la dominación de aristas, tenemos:

- Un conjunto *perfectamente dominante de aristas* de G es un subconjunto $E' \subset E(G)$ tal que toda arista fuera de E' es adyacente a exactamente una arista de E'. Equivalentemente, toda arista en $E \setminus E'$ está dominada por exactamente una arista en E'.
- Un conjunto *eficientemente dominante de aristas* de G es un subconjunto $E' \subset E(G)$ tal que toda arista de $E(G)$ es adyacente a exactamente una arista de E'. Equivalentemente, toda arista está dominada por exactamente una arista en E'. Los conjuntos eficientemente dominantes de aristas también se conocen como *matchings dominantes inducidos (DIM)* pues las aristas E' dominan todas las aristas en G y E' es un matching.

Estos problemas también admiten una versión con pesos. Un *grafo pesado* es un grafo $G = (V, E)$ junto con una función de pesos $w : V \to \mathbb{R}$ en el caso de pesos en los vértices, y/o $w : E \to \mathbb{R}$, en el caso de pesos en las aristas. El peso de un conjunto (de vértices o aristas) es la suma de los pesos de sus elementos. La versión pesada de un problema pregunta por la existencia de un conjunto con mínimo/máximo peso en lugar de mínima/máxima cardinalidad. La versión sin pesos es un caso particular de la versión pesada, ya que es equivalente a poner el mismo peso positivo a todos los elementos.

Organización

En esta tesis abordamos problemas de conjunto dominante con el objetivo de entender las estructuras internas y las propiedades relacionadas con la existencia de soluciones. También estamos interesados en desarrollar algoritmos eficientes para resolver esos problemas.

En el Capítulo 2, las nociones y propiedades de la dominación son discutidas en detalle. También estudiamos el problema de conjunto dominante de aristas restringiendo el dominio a los grafos de intervalos propios y damos un algoritmo lineal para este problema.

El Capítulo 3 está dedicado a la dominación eficiente de aristas y los aspectos combinatorios del problema DIM. Analizamos las subestructuras relacionadas con la existencia de una gran cantidad de DIMs diferentes. Estas estructuras se relacionan con parámetros y subgrafos inducidos, y nos permiten abordar el problema de contar la cantidad de DIMs en diferentes clases de grafos. De esta manera, damos cotas ajustadas para el número de DIMs para las siguientes clases: grafos en general, grafos sin triángulos y grafos conexos. Los grafos extremales, es decir aquellos que tienen la mayor cantidad de DIMs, también son caracterizados para estas clases.

En el Capítulo 4, estudiamos en detalle la dominación perfecta de aristas. Un enfoque usual es dedicir cuál es la complejidad del problema cuando se lo restringe a una cierta clase de grafos. Del manuscrito de Lin y otros [56] se sigue que la complejidad de este problema para los grafos sin garras de máximo grado 3 es NP-completa y la dicotomía se puede completar para grafos de grado acotado cuando la clase de grafos considerada es aquella en la que se prohíbe exactamente un grafo. En este capítulo presentamos propiedades estructurales y algoritmos eficientes (de tiempo lineal) para grafos sin P_5 y grafos cúbicos sin garras .

En el Capítulo 5, centramos nuestra atención en los grafos bien cubiertos ya que el problema de conjunto dominante se resuelve fácilmente para los grafos bien cubiertos de línea y además todo conjunto dominante de aristas en un grafo es un conjunto dominante de vértices en su grafo de línea. Tanto el problema de reconocimiento como el problema de conjunto dominante de vértices para grafos bien cubiertos son, en general, coNP-completo y NP-completo, respectivamente. Nosotros consideramos ambos problemas para algunas subclases de grafos bien cubiertos: grafos bien cubiertos sin $K_{1,4}$, grafos bien cubiertos de comparabilidad y grafos bien cubiertos de co-comparabilidad.

Finalmente, en el Capítulo 6 resumimos y concluimos el trabajo de esta tesis, y presentamos una lista completa de nuestros aportes.

Chapter 1

Introduction

At the present, graph theory is a dynamic field in both theory and applications. The covering problems conforms an increasing field in graph theory. Many of the problems arise from applications and other from theoretical interests and scientific conjectures. Domination theory, in particular edge domination, and its problems are the main object of study in this thesis.

Graphs are useful tools for model problems from very different sources since they serve as a representation of the relations among objects. For instance, a network of cities which are represented by vertices, and connections among them, represented by edges, define a graph. In some contexts, there is also a weight function defined on its vertices and/or its edges. In the above example, the weight of a connection may represent the distance between connected cities. The well-know traveling salesman problem asks for the shortest possible tour, which visits all the cities exactly once. The dominating sets in graphs are natural models for facility location problems in operational research. These problems seek the location of one or more facilities in a way that optimizes a certain objective, such as minimizing transportation cost, providing equitable service to customers and capturing the largest market share. The concept of domination is also applied in coding theory as discussed by Kalbfleisch, Stanton and Horton [52] and Cockayne and Hedetniemi [26]. If one defines a graph, the vertices of which are the n-dimensional vectors with coordinates chosen from $\{1, ..., p\}$, $p > 1$, and two vertices are adjacent if they differ in one coordinate, then the sets of vectors which are (n, p)-covering sets, single error correcting codes, or perfect covering sets are all dominating sets of the graph with determined additional properties.

Scientific conjectures are another source of interesting problems and have had important impact in the development of graph theory. For example, Berge formulated the Strong Perfect Graph Conjecture and introduced perfect graphs in his books [3, 4] in the 1960s. For 40 years, the attempts to solve his conjecture have given rise to powerful methods, important concepts and interest results in different areas of graph theory. Finally, the conjecture was proved in 2006 [25] and now it is referred to as the Strong Perfect Graph Theorem. In domination theory, we have to mention the famous, and still open, Vizing's conjecture formulated in 1963 [83]. The conjecture affirms that the size of a minimum dominating set, called the *dominating number*, of the Cartesian product of two graphs $F \square G$ is at least the product of the dominating numbers of F and G. The approaches to this conjecture include families of graphs for which the conjecture holds, properties of a minimal counterexample, and related conjectures. A good survey of the progress in this field can be found in [15].

From the computational point of view, the problems can be classified according to its hardness following the theory of NP-Completeness formalized by Cook in [27]. Most of the covering problems in graph theory belong to the NP-complete class. The fact that it is not currently possible to solve these problems in a reasonable time lead to different approaches in order to give solutions that may serve for practical purposes. One of these approaches is to restrict the problem domain. Due to many times the input is not arbitrary data, a lot of assumptions over the input instances can be made according to the field where the problem came from; and these restrictions can be formalized by structural properties that every input graph has to satisfy. Several graph classes have been defined following practical purposes and many others arise from theoretical interests. Among the first ones we can name set intersection graphs [33] like interval graphs [11, 55], circular-arc graphs [11, 39], but there are more of them. Other graph classes are defined by restricting the graph parameters like the maximum degree or independent number, and by forbidding some internal structures. A well-organized survey in families of graphs studied in the present time is [11]. In this thesis the algorithms have been designed with special effort in being efficient and robust in order to make them useful for practical purposes.

The theoretical questions about covering problems might be very different. One approach is to bound the number of solutions of one problem. This is related to the algorithmic point of view because yields a natural bound for the exhaustive search. Several related-parameters and variations of domination problems have been defined and extensively studied. Also these kind of analysis is useful to understand the internal structures and, sometimes, develop efficient algorithms to solve related problems.

1.1 Background

Graph theoretical ideas date back to at least 1730's, when Leonard Euler published his paper on the problem of the seven bridges of Köningsberg [6]. In the beginnings the large part of graph theory has been motivated by the study of games and recreational mathematics. Graphs are useful tools for modeling relationships among objects. The usual convention is that objects are represented by vertices (points) and the relationships between them are represented by edges (lines) that join those vertices. In general, any mathematical object involving points and connections between them can be called a graph or a hypergraph. There is no restriction on objects and their relationship, thus graphs can be used in databases, physical networks, signal-flow graphs, social networks or the flow of a computer program among many other applications. Usually, problems from real life can be modeled with graphs satisfying several properties, other graph classes have been defined with theoretical purposes. The advantage of restricting the domains of the problems is that it is easier to work on solution for these restricted representations. An overview of graph classes including intersection graph theory can be found in [11].

Since the concept of domination arises naturally, there are some references to domination-related problem about centuries ago. In 1862 de Jaenisch [29] and in 1892 W.W. Rouse Ball [78] presented different domination problems in chessboard. Domination theory was further developed in the 1950s and 1960s. In 1958 C. Berge wrote a book on graph theory [5], in which he introduced the *coefficient of external stability*, which is now known as the domination number of a graph. In the book of O. Ore [70] the terms *dominating set* and *dominating number* were introduced, but it was after the advances of the brothers Yaglom and Yaglom [86] and the survey paper of Cockayne and Hedetniemi [26] that the notation $\gamma(G)$ was first used for the domination number of a graph. Since then, domination in graph have been extensively studied. The minimum dominating set problem and many of the interesting variations of this problem, were proved to be NP-complete in [38]. Recent surveys of domination theory are [42, 46–48] and several applications of dominating set problems can be found in [73].

In the next section, we give the terminology, definitions and notations used throughout this thesis.

1.2 Notations and Definitions

A *graph* $G = (V, E)$ consists in a (finite) set, denoted by $V(G)$ or V, of *vertices* or *nodes*, and a set $E(G)$ or E of 2-element subset of vertices, called *edges* or *lines*. Edges represent the relation between vertices. The number of vertices, $|V| = n$, is the *order* of the graph, and $|E| = m$ is called its *size*. In the case of *undirected graphs*, edges are represented by unordered pairs of vertices, $e = uv \in E$. For *directed graphs*, each (directed) edge is represented by an ordered pair of vertices, $e = (u, v)$ or $e = \overrightarrow{uv}$. When $e = uv \in E$, we say that vertices u and v are *adjacent* or *neighbor*.In addition, edge e has endpoints u and v, and e is *incident* to u and v. Two edges are *adjacent* if they are incident to a same vertex. Unless we indicate otherwise, all graphs considered in this thesis are undirected.

The set of neighbors of a vertex $v \in V(G)$ is called the *open neighborhood* and it is denoted by $N(v)$, while $N[v] = N(v) \cup \{v\}$ is the *close neighborhood*. The *degree* of a vertex is $\deg(v) = |N(v)|$. The minimum and maximum degree of vertices in $V(G)$ are denoted by $\delta(G)$ and $\Delta(G)$, respectively. A graph where all vertices have the same degree r is called a *r-regular graph*. The 3-regular graph also received the name of *cubic*. For $S \subset V(G)$, $N(S) = \bigcup_{v \in S} N(v) \setminus S$, and $N[S] = \bigcup_{v \in S} N[v]$. An *universal* vertex is a vertex satisfying $N[v] = V(G)$. A subset of non-adjacent vertices is called a *independent* or *stable* set. A *matching* is a subset of non-adjacent edges.

An *induced subgraph* of a graph G is a graph formed by a subset of the vertices of G and all edges of G with both endpoints in those vertices. Given a subset $S \subset V(G)$, we noted by $G[S]$ the subgraph induced by S. An induced subgraph H of G is called *dominating* if the vertices of H, $V(H)$, is a dominating set of G, e.i. every vertex in $V(G) \setminus V(H)$ is adjacent to some vertex in $V(H)$. A graph G is H-free when H is not an induced subgraph of G. A *complete set* in a graph is a subset of vertices that induces a complete graph; that is, every pair of vertices are adjacent. A *clique* is a maximal complete set. The complete graph of order n is denoted by K_n. Graph K_3 is also called *triangle*.

A *path* is a sequence of adjacent edges that connect a sequence of distinct vertices. The number of edges is the length of the path. A graph is *connected* if there is a path joining u and v, for any two vertices u and v; and a *connected component* is a maximal connected subgraph. A *brigde* is an edge e such that its deletion increases the number of connected components of G. The *distance* between two vertices is the length of a shortest path between them, and it is denoted by $dist(u, v)$. Similarly, a *cycle* is a sequence of adjacent vertices such that only the first and the last vertex is the same. The number of edges is the length of the cycle. A path (cycle) is *induced* when no graph edge connects two nonconsecutive vertices of the path (cycle). We respectively denoted by P_n and C_n the path and the cycle of n vertices.

A graph is *bipartite* when its vertex set can be divided into two disjoint, independent sets. Equivalently, G is bipartite when does not contains cycles of odd length. As usual, K_{n_1,n_2} denote the complete bipartite graph; that is, a bipartite graph with vertex set $V_1 \cup V_2$, $|V_1| = n_1$, $|V_2| = n_2$ and every vertex of V_1 is adjacent to every vertex of V_2. Graphs $K_{1,d}$ are called *stars*, and $K_{1,3}$ is also called *claw*.

We follow the notation in [66] to introduce the useful concept of intersection graphs. Given a family of sets $\mathcal{F} = \{S_1, ..., S_n\}$, the *intersection graph of* $\mathcal{F}$, $\Omega(\mathcal{F})$, is the graph formed by vertices $\mathcal{F}$ where S_i and S_j are adjacent if and only if they have nonempty intersection $S_i \cap S_j \neq \emptyset$. A graph is an *intersection graph* if there exists a family $\mathcal{F}$ such that G and $\Omega(\mathcal{F})$ are isomorphic; that is, there is an bijection between their vertex sets preserving adjacency property. Then $\mathcal{F}$ is then called a *representation* of G. Note that a graph can admits different representation set families. Given a graph G, the *line graph* of G, denoted by $L(G)$, is the intersection graph of the edges of G.

The *square of a graph* G, denoted by G^2, is a graph with the same vertex set than G where two vertices are adjacent in G^2 if the distance between them in the graph G is at most 2.

The *complement of a graph* $G = (V, E)$, is a graph denoted by $\bar{G} = (V, \bar{E})$ with the same vertex set than G and where two vertices are adjacent in $\bar{G}$ if and only if they are not adjacent in G.

A *comparability graph* $G = (V, E)$ is an undirected graph such that admits a transitive orientation of its edge set. That is, each edge $e = uv$ is oriented as $\overrightarrow{uv}$ or $\overleftarrow{uv}$, and, in the directed graph defined, whenever edges $\overrightarrow{uv}$ and $\overrightarrow{vw}$ exist, then there also exists edge $\overrightarrow{uw}$.

Graph problems

Given a graph G, $\alpha(G)$ denote the size of a maximum independent set is an independent set; i.e. with the largest number of vertices. Determining $\alpha(G)$ is an NP-hard problem [38]. A graph is *well-covered* if all maximal independent set have the same size. Testing whether a graph contains two maximal independent sets of different sizes is an NP-complete problem; that is, complementarily, testing whether a graph is well-covered is coNP-complete [24, 79].

A *vertex cover* or *cover* of a graph G is a subset of vertices $S \subset V$ such that every edge is incident to at least one vertex of S. Equivalently, if $V \setminus S$ is independent. Therefore, the problem of finding a minimum-size vertex cover is also NP-hard.

We say that a vertex *dominates* itself and every other vertex adjacent to it. A *vertex dominating set* for a graph $G = (V, E)$ is a subset $D \subset V$ such that every vertex of G is dominated by at least one vertex of D. Equivalently, every vertex not in D is adjacent to at least one vertex of D. The *domination number* $\gamma(G)$ is the number of vertices in a smallest dominating set for G. The dominating set problem concerns testing whether $\gamma(G) \leq k$ for a given graph G and input k. Analogously an edge *dominates* itself and every edge adjacent to it. An *edge dominating set* for a graph $G = (V, E)$ is a subset $E' \subset E$ such that every edge in G is dominated by at least one edge from E'. Equivalently, if every edge not in

E' is adjacent to at least one edge of E'. The edge dominating set problem asks for the existence of an edge dominating set satisfying $|D'| \leq k$ for a given graph G and input k. Both problems were probed to be NP-complete in [38] and [40], respectively. Unless it is indicated otherwise 'domination' always refers to vertex domination.

Along this thesis we mention several variations of dominating. We define them below and give more details and references later.

- A *perfect* dominating set for a graph $G = (V, E)$ is a subset $D \subset V$ such that every vertex not in D is adjacent to exactly one vertex of D. Equivalently, every vertex in $V \setminus D$ is dominated by exactly one vertex in D.
- An *efficient* dominating set, or, *independent dominating set*, is a perfect dominating set D such that D is also an independent set. Equivalently, every vertex is dominated by exactly one vertex in D. The *independent domination number*, $i(G)$, is the number of vertices in a smallest independent dominating set for G.

Analogously, for edge domination, we have:

- A *perfect edge dominating set* of G is a subset $E' \subset E(G)$ such that every edge not in E' is adjacent to exactly one edge of E'. Equivalently, every edge in $E \setminus E'$ is dominated by exactly one edge in E'
- A *efficient edge dominating set* of G is a subset $E' \subset E(G)$ such that every edge of $E(G)$ is adjacent to exactly one edge in E'. Equivalently, every edge is dominated by exactly one edge in E'. Efficient edge dominating sets also are known as *dominating induced matching (DIM)* since edges E' dominate all edges in G and E' is a matching.

These problems also admits a weighted version. A *weighted graph* is a graph $G = (V, E)$ and a weight function $w : V \to \mathbb{R}$ in the case of weighted vertices, and/or $w : E \to \mathbb{R}$, in the case of weighted edges. The weight of a set (of vertices or edges) is the sum of the weight of its elements. The weighted version of a problem ask for the existence of a set with minimum/maximum weight instead minimum/maximum cardinality. Unweighted version is a special case of weighted version. It is equivalent to put a same positive weight to every element.

1.3 Overview

In this thesis we aboard dominating set problems with the objective of understand internal structures and properties related to the existence of solutions. In addition, we are interested in developing efficient algorithms to solve these problems.

In Chapter 2, the notions and properties of edge dominating are discussed in detail. We also study the edge dominating set problem restricting the domain to proper interval graphs and we give a linear time algorithm for this problem.

Chapter 3 is dedicated to efficient edge domination and the combinatorial aspects of the DIM problem. We analyze substructures related with the existence of many different DIMs. These structures are related with graph parameters and induced subgraphs, and allows us to approach the problem of counting DIMs in different graph classes. Thus, we give sharp bound for the number of DIMs for the following classes: general graph, triangle-free, connected graphs. The extremal graphs, i.e. graphs having the maximum number of DIMs, are also characterized for these classes.

In Chapter 4, we study in detail perfect edge domination. An usual approach is to decide in which complexity class this dominating problem falls when it is restricted to a graph class. From the manuscript of Lin et al. [56] it follows that the complexity of this problem for claw-free graphs of maximum degree at most three is NP-complete and the dichotomy can be completed for bounded degree graphs where the subgraph classes considered are those that forbid exactly one graph. We present structural properties and efficient (linear-time) algorithms for P_5-free graphs and cubic claw-free graphs in this chapter.

In Chapter 5, we put our attention on well-covered graphs since the vertex dominating set problem is nicely resolved for well-covered line graphs and every edge dominating set for any graph is a vertex dominating set of its line graph. As the recognition problem and the vertex dominating set problem for general well-covered graphs is coNP-complete and NP-complete, respectively. We consider both problems for some subclasses of well-covered graphs: well-covered $K_{1,4}$-free graphs, well-covered comparability graphs and well-covered co-comparability graphs.

Finally, Chapter 6 summarizes and concludes the work in this thesis, and presents a complete list of our contributions.

Chapter 2

Edge Domination

Since it is closely linked with several important graph problems, edge domination has been extensively studied. The problem of finding a minimum edge dominating set in the graph G is equivalent to the problem of finding a minimum vertex dominating set in the line graph $L(G)$. While both the edge dominating set problem and the (vertex) dominating set problem are NP-complete [40], in some ways the problem restricted to line graphs is easier. For instance, minimum dominating set problem is hard to approximate [34], while minimum edge dominating set problem is a constant-factor approximable [22]. Also from the parametrized point of view, minimum dominating set problem most likely is not fixed parameter tractable (it is W[2]-complete [31]), while minimum edge dominating set problem is fixed parameter tractable [35].

As we mentioned before, a possible approach to this problem is to design polynomial-time algorithms for some families of graphs. In this chapter we present a linear time algorithm that solves minimum edge dominating set problem for proper interval graphs. It could be of interest to use the main ideas to yield algorithms for more general classes.

2.1 Preliminaries

We consider simple, finite, undirected graphs $G = (V, E)$. Given an edge $e \in E$, say that e *dominates* itself and every edge sharing a vertex with e. A subset $D \subset E$ is an *edge dominating set* of G if every edge of E is dominated by some edge in D. The edge dominating set problem is to find a minimum edge dominating set of a given graph.

A *matching* is a set of non-adjacent edges (Figure 2.1). A *dominating matching* M is a matching such that M is also a edge dominating set. A matching M is *maximal* if no edge in $E(G) \setminus M$ satisfies $M \cup \{e\}$ is matching. Note that M is a maximal matching if and only if M is a dominating matching. See Figure 2.2.

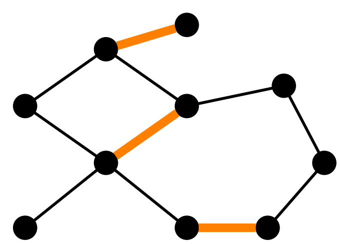

Figure 2.1: A matching in a graph.

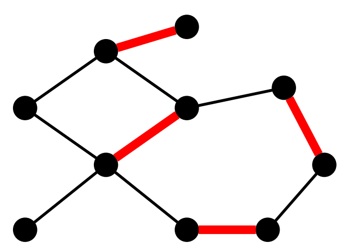

Figure 2.2: A dominating matching or maximal matching in graph.

A *proper interval graph* is the intersection graph of a family of intervals $\mathcal{I} = \{I_k\}_{k=1,\ldots,n}$ where the intervals do not contain each other. Then, $\mathcal{I}$ is called a *proper interval representation* or *model* of the graph. See an example in Figure 2.3. It is well known that, given graph G, it can be recognized in linear time if G is a proper interval graph and, in the positive case, a proper interval representation of G can be obtained in the same time bound [9]. We assume that a proper interval model of G is part of the input for our linear-time algorithm described in Section 2.4.

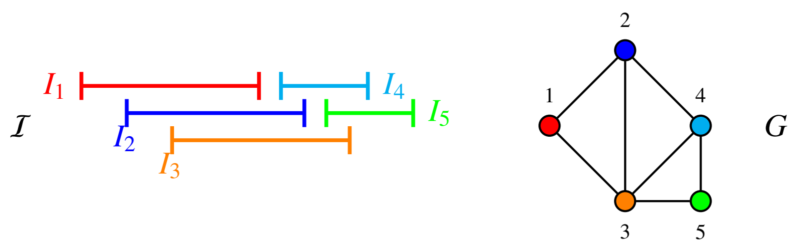

Figure 2.3: A proper interval model $\mathcal{I}$ of the graph G.

2.2 Previous results

The concept of edge domination was introduced by Mitchell and Hedetniemi [67] in 1977, and since then, it has been extensively studied. In 1980, M. Yannakaskis and F. Gravil proved that the problem of finding a minimum edge dominating set is NP-hard even for planar or bipartite graphs with maximum degree 3 [40]. The classes of graphs for which its NP-hardness holds were later refined and extended by Horton and Kilakos to planar bipartite graphs, line and total graphs, perfect claw-free graphs, and planar cubic graphs [51].

Nevertheless, there are some classes for which the problem can be solved in polynomial time. Linear time algorithms were found for trees [67] and block graphs [50], there are polynomial time algorithms for claw-free chordal graphs, locally connected claw-free graphs, line graphs of total graphs, line graphs of chordal graphs [51], bipartite permutation graphs, and cotriangulated graphs [80]. Also, Nagavamsi presented in [69] a polynomial time algorithm for interval graphs.

2.3 Properties related to edge domination

First we present some classical properties that are useful to understand edge domination in general graphs. We include the proofs in order to make this thesis self contained, these and more related properties can be found, for example, in [44].

Lemma 2.1. *A subset $D \subset E(G)$ is a edge domination set if and only if $S = V(G) \setminus V(D)$ is an independent (or stable) set.*

Proof. It follows straightforward from the definition. □

Edge dominating sets in G can be independent sets in $L(G)$ or not. In the positive case, we have dominating matchings of G. Moreover any maximal matching is also a dominating matching and viceversa.

Lemma 2.2. *If $D \subset E(G)$ is an edge dominating set of minimum size then there exists a dominating matching M, such that $|M| = |D|$.*

Proof. Suppose that every dominating matching M satisfies $|M| > |D|$. Then D is not a matching, since whether D is a matching, then $|M| = |D|$. Let D be a minimum edge dominating set with the maximum number of independent edges and let $x, y, z \in V(G)$ be vertices such that $xy, yz \in D$.
There exists an edge such that only xy dominates, otherwise xy can be removed from D contradicting the minimality of D. Hence, let ax be this edge. Therefore, $(D \setminus \{xy\}) \cup \{ax\}$ is a minimum edge dominating set with more independent edges than D, contradicting the choice of D. □

Since any dominating matching is, in particular, an edge dominating set, we have $|M| \geq |D|$ for any D minimum edge dominating set and M a dominating matching. And by Lemma 2.2, there exists a matching such that $|M| = |D|$. Therefore, the size of a minimum edge dominating set equals the size of a minimum dominating matching. And, equivalently, any minimum maximal matching is a minimum edge dominating set.

The following corollary is a consequence of this observation and Lemmas 2.1 and 2.2.

Corollary 2.1. *Any maximal matching in a complete graph G is a minimum edge dominating set of G.*

2.4 Algorithm for Proper Interval Graphs

In this section, we describe an efficient algorithm for finding a minimum edge domination set of a proper interval graph. The greedy Algorithm 1 computes a minimum dominating matching and, by Lemma 2.2, it is also a minimum edge dominating set.

We present a linear-time algorithm to find a minimum edge dominating set of a proper interval graph G. The input is a proper interval model $\mathcal{I}$ of G and the output is (M, k) where M is a minimum dominating matching and $k = |M|$. See in Figure 2.4 an example of Algorithm 1 running on a proper interval model of the graph in Figure 2.5.

In the process the algorithm will select some edges to belong to the dominating matching from the neighborhood of a fixed vertex. We introduce additional parameters in order to make the algorithm easy to follow. Suppose that the intervals of the proper interval model are ordered by their occurrence in the real line. For every interval I_i, the number $\rho(i)$ indicates the last interval intersecting with it, this is $I_i \cap I_{\rho(i)} \neq \emptyset$ and $I_i \cap I_{\rho(i)+1} = \emptyset$. Note that $i \leq \rho(i)$ and vertices $i, i+1, ..., \rho(i)$ induce a complete subgraph. Usually the algorithm will select edges with the form $I_i I_{i+1}$ but if vertex $i+1$ has been matched previously the procedure is different. The parameter $\mu(j)$, for $j = 1, ..., n$, will indicate if vertex j is matched. (We let parameter μ equals 1 only for those vertices j matched with $j-2$, in other words, when the algorithm selects edge $I_{j-2} I_j$.)

Algorithm 1 Minimum edge dominating set for proper interval graphs

Input: A proper interval model $\mathcal{I} = \{I_k\}_{k=1,\dots,n}$ of G

Output: (M, k) where M is a minimum dominating matching and $k = |M|$

1) compute each value of the function $\rho : \{1, \dots, n\} \to \{1, \dots, n\}$ where $\rho(i) = max\{j / I_i \cap I_j \neq \emptyset\}$ and the array $\mu[1..n]$ with initial values $\mu[i] = 0$ for $1 \leq i \leq n$,
2) $M := \emptyset$, $k := 0$, $i := 1$
3) **if** $\rho(i) - i - \mu(i+1) + 1$ is odd **then**
   $M := M \cup \{I_{\rho(i)}I_{\rho(i)-1}, I_{\rho(i)-2}I_{\rho(i)-3}, \dots, I_{i+2+\mu(i+1)}I_{i+1+\mu(i+1)}\}$, $k := k + \frac{\rho(i)-i-\mu(i+1)}{2}$, $i := \rho(i) + 1$ and GOTO (10)
4) **end if**
5) **if** $\rho(\rho(i)) = \rho(i)$ **then**
   $M := M \cup \{I_{\rho(i)}I_{\rho(i)-1}, I_{\rho(i)-2}I_{\rho(i)-3}, \dots, I_{i+3+\mu(i+1)}I_{i+2+\mu(i+1)}, I_{i+1+\mu(i+1)}I_i\}$, $k := k + \frac{\rho(i)-i-\mu(i+1)+1}{2}$, $i := \rho(i) + 1$ and GOTO (10)
6) **end if**
7) **if** $\rho(i) + 2 \leq n$ and $\rho(i) + 2 \leq \rho(\rho(i))$ **then**
   $M := M \cup \{I_{\rho(i)+2}I_{\rho(i)}, I_{\rho(i)-1}I_{\rho(i)-2}, \dots, I_{i+2+\mu(i+1)}I_{i+1+\mu(i+1)}\}$, $k := k + \frac{\rho(i)-i-\mu(i+1)+1}{2}$, $\mu(\rho(i) + 2) := 1$, $i := \rho(i) + 1$
8) **else**
   $M := M \cup \{I_{\rho(i)+1}I_{\rho(i)}, I_{\rho(i)-1}I_{\rho(i)-2}, \dots, I_{i+2+\mu(i+1)}I_{i+1+\mu(i+1)}\}$, $k := k + \frac{\rho(i)-i-\mu(i+1)+1}{2}$, $i := \rho(i) + 2$
9) **end if**
10) **if** $i < n$ **then**
    GOTO (3)
11) **end if**
12) **return** (M, k)

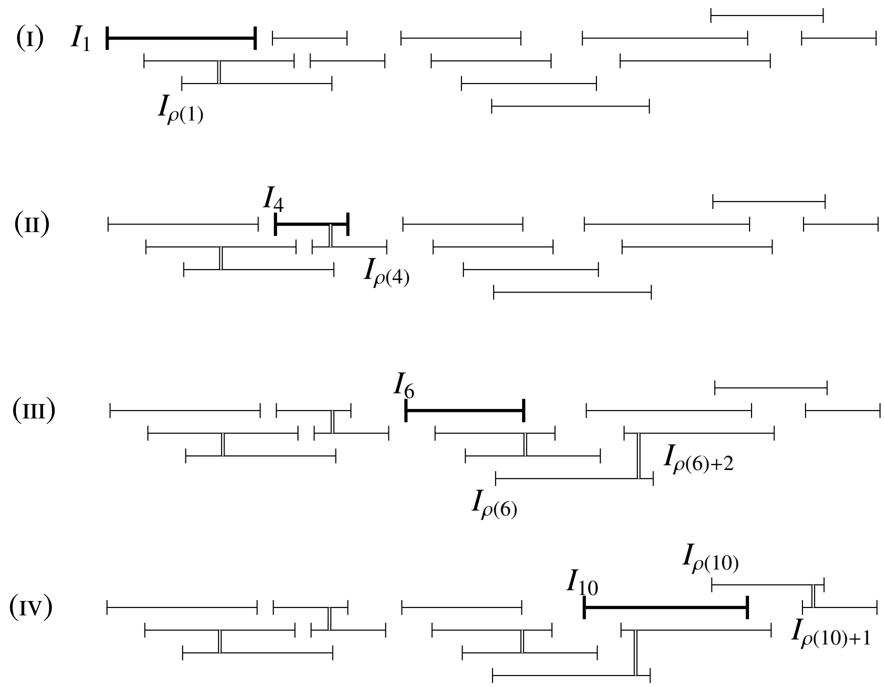

Figure 2.4: Different steps of Algorithm 1 while running for a proper interval model of graph G in Figure 2.5. The output is ($M = \{I_2I_3, I_4I_5, I_7I_8, I_9I_{11}, I_{12}I_{13}\}, k = 5$).

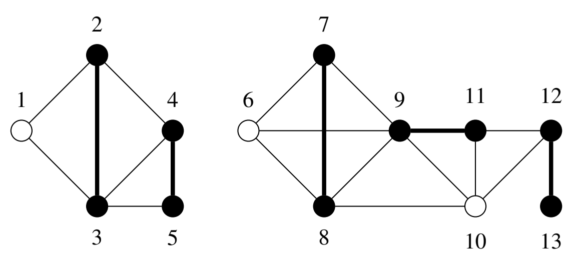

Figure 2.5: Graph G and the output of Algorithm 1. $M = \{I_2I_3, I_4I_5, I_7I_8, I_9I_{11}, I_{12}I_{13}\}$ is a minimum dominating matching.

We now proceed to prove the correctness and efficiency of the algorithm. Note that the input can be a disconnected graph, and the algorithm ends because counter i is always increasing. Let M be the output of Algorithm 1, $V(M)$ the endpoints of edges in M and $S = V(G) \setminus V(M)$.

Lemma 2.3. *M is a matching, S is an independent set, and if $j \in S$ then, in a certain moment, the algorithm analyze interval I_j, i.e. in a certain moment $i = j$.*

Proof. When the algorithm analyzes an interval I_i, all intervals in $\{I_i, I_{i+1} ..., I_{\rho(i)}\}$ will be covered by M, the only exception can be I_i and then $i \in S$. If, in addition, $\rho(i) + 1 \in V(M)$ next interval to be analyzed is $I_{\rho(i)+2}$, otherwise it is $I_{\rho(i)+1}$. Since $\mathcal{I}$ is a proper interval model, S results an independent set. □

Lemma 2.4. *If the algorithm analyze interval I_i then (i) $I_i \in S$ or (ii) $\rho(i) - i - \mu(i + 1) + 1$ is even, $\rho(\rho(i)) = \rho(i)$, and there is no edge between $G[1, ..., \rho(i)]$ and $G[\rho(i) + 1, ..., n]$.*

Proof. When the algorithm analyzes an interval I_i, vertex i is covered by M only if condition of Step 5 is met. This is, $\rho(i) - i - \mu(i + 1) + 1$ is even and $\rho(\rho(i)) = \rho(i)$. Therefore, no vertex from $\{1, ..., \rho(i)\}$ is adjacent to any vertex in $\{\rho(i) + 1, ..., n\}$. □

Lemma 2.5. *If neither condition of Steps 3, 5 nor 7 is met, then there is no edge between $G[1, ..., \rho(i)]$ and $G[\rho(i) + 2, ..., n]$.*

Proof. When the algorithm analyzes an interval I_i, and neither condition of Steps 3, 5 nor 7 is met, then $\rho(i) - i - \mu(i + 1) + 1$ is even, and $\rho(\rho(i)) > \rho(i)$. This implies that there exists vertex $\rho(i) + 1$. If $n = \rho(i) + 1$, the statement is true. If there exists $\rho(i) + 2$ then $\rho(i)+2 > \rho(\rho(i))$. This implies that no vertex from $\{1, ..., \rho(i)\}$ can be adjacent to any vertex in $\{\rho(i) + 2, ..., n\}$. □

Lemma 2.6. *Given a graph G and I_j the first interval of any connected component C, then (i) $I_j \in S$ or (ii) C is complete and the number of its vertices is even.*

Proof. It is easy to see that in one moment of the execution of the algorithm we consider j as i because an interval I_k is skipped only if it intersects with some previous interval I_t, $t < k$. Therefore, by Lemma 2.4, if $I_j \notin S$ then $\rho(j) - j - \mu(j+1) + 1$ is even and $\rho(\rho(j)) = \rho(j)$. Since I_j is the first interval of C, $\mu(j+1) = 0$ and the vertices in C are $\{j, ..., \rho(j)\}$. Consequently, C is complete and $|C|$ is even. Moreover, the output of the algorithm contains a minimum dominating matching of C consisting of $\{I_j I_{j+1}, ..., I_{\rho(j)-1} I_{\rho(j)}\}$. □

Theorem 2.1. *The Algorithm 1 computes a minimum dominating matching of G.*

Proof. By Lemma 2.3, S is an independent set and M is a dominating matching of G. Let be I_i the first interval added to S which implies that vertices $\{1, ..., i-1\}$ induce complete components of G and each one has an even number of vertices by previous Lemma 2.6. Clearly, any maximal matching (minimum dominating matching) has exactly $\frac{i-1}{2}$ edges in these components. Without loss of generality, we can assume that vertex $1 \in S$.

Suppose that there exists a dominating matching M^* with $|M^*| < |M|$, and then $|S^*| > |S|$. We can compare $s^*(k) = |S^* \cap \{1, .., k\}|$ and $s(k) = |S \cap \{1, ..., k\}|$ for $1 \leq k \leq n$. It is clear that $s^*(k) \leq s^*(k+1)$ and $s(k) \leq s(k+1)$ for any $k \in \{1, ..., n-1\}$. As $s(1) = 1$ and $s^*(1) \leq 1$, then $s(n) = |S|$ and $s^*(n) = |S^*| > |S|$, there is a smallest j that verifies $s^*(j) > s(j)$ and $j \in S^*$. Clearly $s^*(j) = s(j) + 1$. Let i' and i'^* be the last vertices in $S \cap \{1, .., j-1\}$ and $S^* \cap \{1, .., j-1\}$, respectively. We have $i' \leq i'^*$, $\rho(i'^*) < j$ (because i^* and j are non-adjacent vertices, both of them belong to S^*) and then $\rho(i') < j$.

Every interval I_i analyzed by the algorithm with $i' < i \leq j$ satisfies $I_i \notin S$ and there is at least one. Suppose that i is the first vertex among them such that $j \leq \rho(i)$. By Lemma 2.4 we have that $G[1, ..., \rho(i)]$ is disconnect with $G[\rho(i)+1, ..., n]$. The number of edges in $M \cap E(G[1, ..., \rho(i)])$ is exactly $\frac{\rho(i)-s(j)}{2}$ because any k, $i' \leq k \leq j$, satisfies $k \notin S$, and the number of edges in $M^* \cap E(G[1, ..., \rho(i)])$ is exactly $\frac{\rho(i)-s^*(j)}{2}$ because any k, $j \leq k \leq \rho(i)$,

$k \notin S^*$, satisfies that I_k and I_j intersects. This is an absurd since $s^*(j) = s(j) + 1$ and the number of edges should be an integer number.

The other case is that there exists a vertex i, $i' < i \leq j$, such that $\rho(i) + 1 = j$ and the next vertex analyzed by the algorithm is $\rho(i) + 2$. This only can occur if neither of conditions of Steps 3, 5, and 7 was fulfilled for vertex i and then, by Lemma 2.5, there is no edge between $G[1, ..., j-1]$ and $G[j+1, ..., n]$. Therefore, the number of edges in $M \cap E(G[1, ..., j])$ and $M^* \cap E(G[1, ..., j])$ is exactly $\frac{j-s(j)}{2}$ and $\frac{j-s^*(j)}{2}$, respectively. An absurd because $s^*(j) = s(j) + 1$ and the number of edges should be an integer number.

We conclude that M is a minimum dominating matching of G. □

It not difficult to see that Algorithm 1 runs in linear time. Then we have proved following theorem.

Theorem 2.2. *There exists a linear-time algorithm that solves edge dominating set problem for proper interval graphs.*

Resumen del Capítulo 2

Como mencionamos anteriormente, una estrategia para tratar un problema NP-completo como lo es el problema del conjunto dominante de aristas [40] es diseñar algoritmos de tiempo polinomial para algunas familias de grafos. En este capítulo usamos esta estrategia para resolver el problema en la clase de los grafos de intervalos propios.

En la Sección 2.1 damos las definiciones necesarias.

En un grafo no dirigido $G = (V, E)$, una arista $e \in E$ se *domina* a sí misma y a toda arista que comparte un vértice con ella. Un subconjunto $D \subset E$ es un *conjunto dominante de aristas* si toda arista de E está dominada por alguna arista en D. El problema del conjunto dominante consiste en encontrar un conjunto dominante de tamaño mínimo para un grafo dado.

Un *grafo de intervalos propios* es el grafo de intersección de una familia de intervalos $\mathcal{I} = \{I_k\}_{k=1,\ldots,n}$ donde los intervalos no se contienen entre sí. En ese caso, la familia $\mathcal{I}$ se llama una *representación* o *modelo* del grafo. Se sabe que, dado un grafo G, se puede reconocer en tiempo linear si G es un grafo de intervalos propios y, en ese caso, se puede obtener una representación de G en intervalos propios en el mismo tiempo [9].

En la Sección 2.3 damos algunas propiedades clásicas que son útiles para entender la dominación de aristas en la clase general de grafos. Entre ellas usaremos especialmente que el tamaño de un conjunto dominante de aristas mínimo equivale al tamaño de un matching maximal mínimo.

En la Sección 2.4 damos un algoritmo para encontrar un conjunto dominante de aristas de tamaño mínimo en un grafo de intervalos propios. El Algoritmo 1 devuelve un matching dominante de mínimo tamaño y, por el Lema 2.2, es también un conjunto dominante de aristas mínimo. Podemos asumir que un modelo de intervalos propios del grafo es parte de la entrada de nuestro algoritmo. Entonces obtenemos que el problema de dominación de aristas es lineal para los grafos de intervalos propios (Teorema 2.2).

Chapter 3

Efficient Edge Domination

Packing and covering problems in graphs and their relationships belong to the most fundamental topics in combinatorics, and graph algorithms and have a wide spectrum of applications in computer science, operations research and many other fields. Recently, there has been an increasing interest in problems combining packing and covering properties. Among them, there are the efficient edge dominating set problem (also known as dominating induced matching or DIM for short) and the perfect edge dominating set problem (that will be discussed in the next chapter). Studies about dominating induced matchings and some applications related to coding theory, network routing, and resource allocation can be found in [43, 62].

In the present chapter we give a combinatorial approach to the efficient edge dominating set problem. We give tight upper bounds on the maximum possible number of DIMs for a graph G that is either arbitrary, or triangle-free, or connected. Furthermore, we characterize all extremal graphs for these bounds. Our results imply that if G is a graph of order n and $\mu(G)$ is the number of DIMs in the graph, then $\mu(G) \leq 3^{\frac{n}{3}}$; $\mu(G) \leq 4^{\frac{n}{5}}$ provided G is triangle-free; and $\mu(G) \leq 4^{\frac{n-1}{5}}$ provided $n \geq 9$ and G is connected.

The structure of this chapter is as follows. First we define the problem in Section 3.1 and dedicate Section 3.2 to present the state of the art. In Section 3.3 we present lemmas and important observations to understand the problem of counting DIMs. In Section 3.4 we state and prove sharp bounds of the number of DIMs and determine all extremal graphs when the problem is restricted to the classes of general graphs, triangle-free graphs and connected graphs.

3.1 Preliminaries

Recall that an edge in $E(G)$ *dominates* itself and every edge sharing a vertex with it. A subset $E' \subset E(G)$ is *dominating* if every edge of G is dominated by at least one edge of E'. A subset $E' \subset E(G)$ is an *efficient dominating set* if every edge of G is dominated by exactly one edge of E'. A *matching* is a subset of non-adjacent edges. An *induced matching* is a set of edges $M \subset E(G)$ such that every edge of G is adjacent to at most one edge in M (See Figures 3.1 and 3.2).

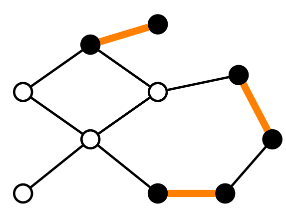

Figure 3.1: A matching in a graph. This matching is not induced.

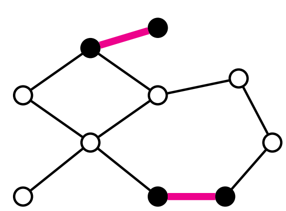

Figure 3.2: An induced matching in a graph.

A *dominating induced matching* or DIM, for short, is a subset of edges $M \subset E(G)$ which is both dominating and an induced matching. In consecuence, an efficient edge dominating set is equivalent to a dominating induced matching (see Figure 3.3).

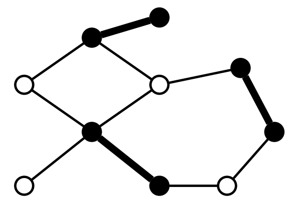

Figure 3.3: A DIM in a graph. Every edge is dominated by exactly one edge in the DIM.

Not every graph has a DIM; for example C_4 or any cycle C_n with $n \neq 3k$ does not admit DIMs (see Figure 3.4). The problem of deciding whether a given graph admits it is known in the literature as the *efficient edge domination or DIM problem*; and it is NP-complete [43]. Given a weight function $w : E(G) \to \mathbb{R}$, the *weighted version* of the DIM problem is to find a DIM with minimum sum of the weight of its edges amongs all DIMs, if any. We denote by $\mu(G)$ the number of dominating induced matchings of G. The problem of compute $\mu(G)$ is the *counting version* of the DIM problem.

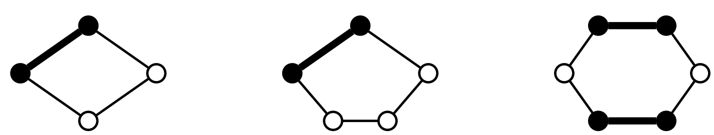

Figure 3.4: Induced matching in cycles C_4, C_5 and C_6. Note that only C_6 admits a DIM.

In this chapter we use an alternative definition [19] of the DIM problem. It is determine if exists a partition of the vertex set into two disjoint subsets; one subset, corresponding to the white vertices, induces an independent set, while the other subset, called the black vertices, induces an 1-regular graph. Thus, black vertices will induce a matching.

It is not difficult to see that a DIM is also a maximum induced matching [17].

Theorem 3.1. *[17] If M is a DIM of a graph G then M is a maximum induced matching in G. Consequently, all DIMs in G have the same cardinality.*

3.2 Previous results

The algorithmic questions related to dominating induced matchings have been studied in great detail. As we mentioned before, the problem of deciding if a given graph admits a DIM is NP-complete [43], and it remains NP-complete even when restricted to planar bipartite graphs of maximum degree 3 [10] or k-regular graphs, for $k \geq 3$ [17]. On the other hand, the DIM problem becomes polynomial-time solvable for several graph classes such as regular graphs [17], claw-free graphs [18], long claw-free graphs [49], graphs with bounded clique-width [18], hole-free graphs [10], convex graphs [53], dually-chordal graphs [12], weakly chordal graphs [12], bipartite permutation graphs [64], AT-free graphs [12], interval-filament graphs [12]. About the weighted version of the efficient edge dominating set problem we have to mention the existence of polynomial-time algorithms for chordal graphs [63], generalized series-parallel graphs [63], P_8-free graphs [14]. Exact algorithms for the general case are also developed, the more recent results in this direction are: an algorithm for the unweighted DIM problem that runs in $1.1467^n n^{O(1)}$-time and polynomial space [85], and an $O^*(1.1939^n)$ algorithm that solves the weighted DIM problem [57].

There is a relationship between DIMs in a graph G and maximum weighted independent sets (MWIS) in the square of the line graph $L(G)$. In consequence, the problem of counting DIMs in G can be approached by counting MWIS in $L^2(G)$. In [28], an $O^*(1.3247^n)$ algorithm for the counting version of MWIS is given, leading to an $O^*(1.3247^m)$ time algorithm to count the numbers of DIMs in a arbitrary graph.

3.3 Colorings associated to a DIM

Throughout this section we consider graphs that are not necessary connected. Clearly a DIM of G is union of DIMs of its connected components. Then the number of DIMs in a non connected graph is the product of the number of DIMs of its connected components.

Assigning one of two possible colors to vertices of G is called a *coloring* of G. A coloring, noted by $(G; B, W)$, splits the set of vertices of G into 3 disjoint subsets, the black vertices B, the white vertices W and the uncolored vertices $V \setminus (B \cup W)$. A coloring is *total* if all vertices of G have colors assigned, otherwise it is *partial*.

Let B and W be disjoint subsets of the vertex set $V(G)$, we say that a dominating induced matching M of G is *compatible with the coloring* $(G; B, W)$ if $B \subseteq V(M)$ and $W \cap V(M) = \emptyset$. Let $\mu(G; B, W)$ denote the number of dominating induced matchings of G that are compatible with $(G; B, W)$.

By the definition of DIMs, we have

$$\mu(G; B, W) > 0 \Rightarrow G[B] \text{ has maximum degree at most 1 and } W \text{ is independent.} \quad (3.1)$$

Note that if $V(G) \setminus (B \cup W)$ has at most n elements, then $\mu(G; B, W)$ is an integer not greater than 2^n. This implies that for a class $\mathcal{G}$ of graphs and a non-negative integer n, the maximum

$$\begin{aligned} s_{\mathcal{G}}(n) \quad = \quad & \max\{\mu(G; B, W) : \; G \text{ is a graph in } \mathcal{G}, B \text{ and } W \text{ are disjoint subsets of } V(G), \\ & B \cup W \neq \emptyset, \text{ and } |V(G) \setminus (B \cup W)| \leq n\} \end{aligned}$$

is well-defined and finite, even though the maximum is possibly taken over infinitely many graphs. Note that a total coloring can be compatible with at most one DIM, then $s_{\mathcal{G}}(0) \leq 1$. In addition, if $\mathcal{G}$ contains a non-empty graph that has a DIM, then $s_{\mathcal{G}}(n) \geq 1$.

We now explain some lemmas for bounding the number of DIMs that are compatible with a coloring $(G; B, W)$

Lemma 3.1. *If $\mathcal{C}$ is the class of connected $\{C_3, C_4\}$-free graphs of minimum degree at least 2, then $s_{\mathcal{C}}(n) = 1$ for $n \in \{0, 1, 2, 3\}$ and $s_{\mathcal{C}}(n) \leq s_{\mathcal{C}}(n-4) + s_{\mathcal{C}}(n-2)$ for every integer n with $n \geq 4$.*

Proof. We prove the statement by induction on n. For $n = 0$, the statement follows from the above observations using that C_6 has DIMs, $\mu(C_6) > 0$ and $C_6 \in \mathcal{C}$. Now let $n \geq 1$. Clearly, $s_{\mathcal{C}}(n) \geq 1$ and $s_{\mathcal{C}}(n-1) \leq s_{\mathcal{C}}(n)$. Hence, in view of the desired statement, we may assume that $s_{\mathcal{C}}(n) \geq 2$. Let $(G; B, W)$ be a maximizer in the definition of $s_{\mathcal{C}}(n)$, that is, $s_{\mathcal{C}}(n) = \mu(G; B, W)$. Since $s_{\mathcal{C}}(n) \geq 2$, the set $B \cup W$ is a proper non-empty subset of $V(G)$. Since G is connected, there is an edge uv of G such that $u \in B \cup W$ and $v \in V(G) \setminus (B \cup W)$.

If $u \in W$, then $s_{\mathcal{C}}(n) = \mu(G; B, W) \overset{(3.1)}{=} \mu(G; B \cup \{v\}, W) \leq s_{\mathcal{C}}(n-1)$. If $u \in B$ and u has a neighbor in B, then $s_{\mathcal{C}}(n) = \mu(G; B, W) \overset{(3.1)}{=} \mu(G; B, W \cup \{v\}) \leq s_{\mathcal{C}}(n-1)$. If $u \in B$ and all neighbors of u distinct from v belong to W, then $s_{\mathcal{C}}(n) = \mu(G; B, W) \overset{(3.1)}{=} \mu(G; B \cup \{v\}, W) \leq s_{\mathcal{C}}(n-1)$. In all three cases, we obtain $s_{\mathcal{C}}(n) \leq s_{\mathcal{C}}(n-1)$. By induction, if $n-1 \leq 3$, then $1 \leq s_{\mathcal{C}}(n) \leq s_{\mathcal{C}}(n-1) = 1$, and if $n-1 \geq 4$, then $s_{\mathcal{C}}(n) \leq s_{\mathcal{C}}(n-1) \leq s_{\mathcal{C}}(n-5) + s_{\mathcal{C}}(n-3) \leq s_{\mathcal{C}}(n-4) + s_{\mathcal{C}}(n-2)$.

Hence we may assume that u belongs to B and that u has a neighbor w in $V(G) \setminus (B \cup W)$ that is distinct from v. Since G is of minimum degree at least 2, the vertex v has a neighbor v' distinct from u and the vertex w has a neighbor w' distinct from u. Since G is $\{C_3, C_4\}$-free, the vertices v, v', w, and w' are all distinct.

If $v' \in W$, then $s_{\mathcal{C}}(n) = \mu(G; B, W) \overset{(3.1)}{=} \mu(G; B \cup \{v\}, W) \leq s_{\mathcal{C}}(n-1)$. If $v' \in B$, then $s_{\mathcal{C}}(n) = \mu(G; B, W) \overset{(3.1)}{=} \mu(G; B, W \cup \{v\}) \leq s_{\mathcal{C}}(n-1)$. Again we obtain $s_{\mathcal{C}}(n) \leq s_{\mathcal{C}}(n-1)$ and can argue as above.

Hence we may assume that v' and w' belong to $V(G) \setminus (B \cup W)$, which implies $n \geq 4$.

Now

$$\begin{aligned} s_C(n) &= \mu(G; B, W) \\ &= \mu(G; B \cup \{v\}, W) + \mu(G; B, W \cup \{v\}) \\ &\overset{(3.1)}{=} \mu(G; B \cup \{v, w'\}, W \cup \{v', w\}) + \mu(G; B \cup \{v'\}, W \cup \{v\}) \\ &\leq s_C(n-4) + s_C(n-2), \end{aligned}$$

which completes the proof. □

If $F(n)$ denotes the n-th Fibonacci number, that is, $F(0) = 0$, $F(1) = 1$, and $F(n) = F(n-2) + F(n-1)$ for every integer n with $n \geq 2$, then Lemma 3.1 immediately implies

$$\max\{s_C(2n), s_C(2n+1)\} \leq F(n+1) \tag{3.2}$$

for every non-negative integer n.

Lemma 3.2. *If G is a $\{C_3, C_4\}$-free graph of order n and minimum degree at least 2, then $\mu(G) < 0.928 \cdot \phi^{\frac{n}{2}}$ where $\phi = \frac{1+\sqrt{5}}{2}$.*

Proof. First, we assume that G is connected. Note that $n \geq 5$. If $n = 5$, then $G = C_5$ and $\mu(G) = 0$. Hence let $n \geq 6$. Let u be a vertex of G. Since u has at least 2 neighbors, we obtain $\mu(G) = \mu(G; \emptyset, \{u\}) + \mu(G; \{u\}, \emptyset) \overset{(3.1)}{=} \mu(G; N(u), \{u\}) + \mu(G; \{u\}, \emptyset) \leq s_C(n-3) + s_C(n-1)$. If n is odd, then $\mu(G) \leq s_C(n-3) + s_C(n-1) \overset{(3.2)}{\leq} F\left(\frac{n-1}{2}\right) + F\left(\frac{n+1}{2}\right) = F\left(\frac{n+3}{2}\right)$. If n is even, then $\mu(G) \leq s_C(n-3) + s_C(n-1) \overset{(3.2)}{\leq} F\left(\frac{n-2}{2}\right) + F\left(\frac{n}{2}\right) = F\left(\frac{n+2}{2}\right)$. Using $\phi^{-2} + \phi^{-1} = 1$ and $\max\left\{F(4) \cdot \phi^{-\frac{6}{2}}, F(5) \cdot \phi^{-\frac{7}{2}}, F(6) \cdot \phi^{-\frac{9}{2}}\right\} < 0.928$, it follows easily by induction on n that for $n \geq 6$, we have

$$\left.\begin{array}{ll} F\left(\frac{n+3}{2}\right) & \text{, if } n \text{ is odd and} \\ F\left(\frac{n+2}{2}\right) & \text{, if } n \text{ is even} \end{array}\right\} < 0.928 \cdot \phi^{\frac{n}{2}}$$

and hence $\mu(G) < 0.928 \cdot \phi^{\frac{n}{2}}$.

If G has components $G_1, ..., G_k$ of orders $n_1, ..., n_k$, respectively, then $\mu(G) \leq \prod_{i=1}^{k} \mu(G_i) < 0.928^k \cdot \phi^{\frac{n_1+...+n_k}{2}} \leq 0.928 \cdot \phi^{\frac{n}{2}}$, which completes the proof. □

3.4 Bounds for the maximum number of DIMs

We proceed to state and prove tight bounds to $\mu(G)$ for the general class of graphs, triangle-free and connected graphs, in the subsequent Subsections. The general structure of all three proofs is very similar.

3.4.1 General graphs

Theorem 3.2. *If G is a graph of order n, then $\mu(G) \leq f(n)$ where*

$$f(n) = \begin{cases} 1 & \text{, if } n \leq 2, \\ 3^{\frac{n}{3}} & \text{, if } n \geq 3 \text{ and } n \equiv 0 \text{ mod } 3, \\ 3^{\frac{n-1}{3}} & \text{, if } n \geq 4 \text{ and } n \equiv 1 \text{ mod } 3, \text{ and} \\ 4 \cdot 3^{\frac{n-5}{3}} & \text{, if } n \geq 5 \text{ and } n \equiv 2 \text{ mod } 3. \end{cases}$$

Furthermore, if the graph G of order n with $n \geq 3$ is such that $\mu(G) = f(n)$, then $G \in \mathcal{F}$ where

$$\begin{aligned} \mathcal{F} &= \left\{\frac{n}{3}K_3 : n \geq 3 \text{ and } n \equiv 0 \text{ mod } 3\right\} \\ &\cup \left\{K_1 \cup \frac{n-1}{3}K_3 : n \geq 4 \text{ and } n \equiv 1 \text{ mod } 3\right\} \\ &\cup \left\{K_{1,3} \cup \frac{n-4}{3}K_3 : n \geq 4 \text{ and } n \equiv 1 \text{ mod } 3\right\} \\ &\cup \left\{K_{1,4} \cup \frac{n-5}{3}K_3 : n \geq 5 \text{ and } n \equiv 2 \text{ mod } 3\right\}. \end{aligned}$$

In Figure 3.5 we show the extremal graphs for the general class of graphs.

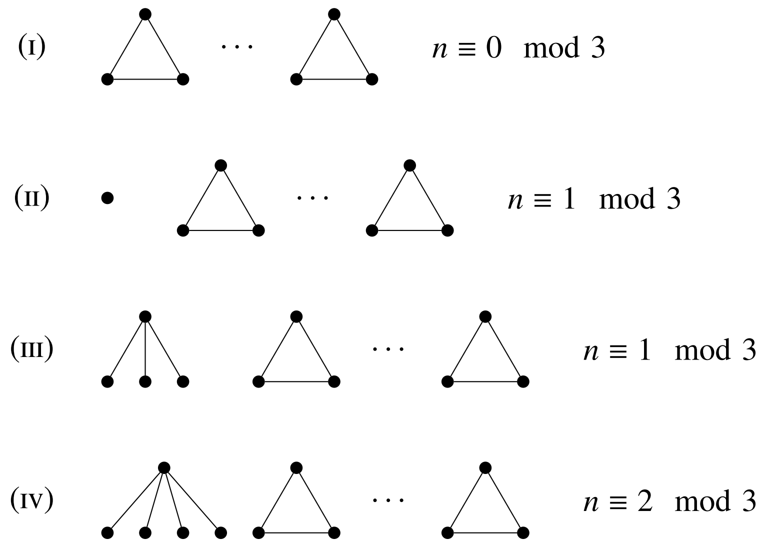

Figure 3.5: Family $\mathcal{F}$. Extremal graphs for the general class of graphs.

Proof of Theorem 3.2

Let G be a graph of order n and size m. We prove, by induction on $n+m$, that $\mu(G) \leq f(n)$, and for $n \geq 3$, $\mu(G) = f(n)$ if and only if G belongs to $\mathcal{F}$. Since the result is easily verified for $n \leq 5$, we assume now that $n \geq 6$. We establish a series of claims concerning properties that G can be assumed to have if G has maximum $\mu(G)$ among all graphs of order n.

Claim 1. *Every edge of G belongs to some DIM of G.*

Proof of Claim 1. If G contains an edge e such that no DIM of G contains e, then every DIM of G is a DIM of $G - e$ and, by induction, $\mu(G) \leq \mu(G-e) \leq f(n)$. If $\mu(G) = f(n)$, then $\mu(G-e) = f(n)$ and hence, by induction, $G - e \in \mathcal{F}$. It is easily verified that adding any edge to a graph H in $\mathcal{F}$ results in a graph with strictly less DIMs than H. Therefore, $\mu(G) < \mu(G-e)$, which is the contradiction $\mu(G) < f(n)$. □

Since no DIM of G can contain an edge that belongs to a cycle of length 4, Claim 1 implies that G has no such cycle.

Following claims assume that G is an extremal graph and $G \notin \mathcal{F}$.

Claim 2. *The graph G is triangle-free.*

Proof of Claim 2. Let $T = G[xyz]$ be a triangle in G. Since every DIM of G contains exactly one of the three edges of T, no DIM of G contains an edge between a vertex in $V(T)$ and a vertex in $V(G) \setminus V(T)$. By Claim 1, this implies that T is a component of G. Now, by induction, $\mu(G) = 3 \cdot \mu(G - V(T)) \leq 3 \cdot f(n-3) = f(n)$. Furthermore, if $\mu(G) = f(n)$, then $\mu(G - V(T)) = f(n-3)$ and hence, by induction, $G - V(T) \in \mathcal{F}$. Since G is the disjoint union of a triangle and $G - V(T)$, we obtain $G \in \mathcal{F}$. □

Claim 3. *The graph G has no isolated vertex.*

Proof of Claim 3. If u is an isolated vertex of G, then every DIM of G is a DIM of $G - u$. Therefore, by induction, $\mu(G) \leq \mu(G - u) \leq f(n-1) \leq f(n)$. If $\mu(G) = f(n)$, then $f(n-1) = f(n)$, which implies that $n \equiv 1 \bmod 3$. Furthermore, $\mu(G - u) = f(n-1)$ and hence, by induction, $G - u = \frac{n-1}{3}K_3$. Now $G = K_1 \cup \frac{n-1}{3}K_3 \in \mathcal{F}$. □

Claim 4. *The graph G has minimum degree at least* 2*.*

Proof of Claim 4. By Claim 3, the graph G has no isolated vertex. If u is a vertex of degree 1 and v is the unique neighbor of u in G, then every DIM of G contains an edge incident with v. Hence no DIM contains an edge between a vertex in $N[v]$ and $V(G) \setminus N[v]$. By Claim 1 and Claim 2, the closed neighborhood $N[v]$ of v in G is the vertex set of a component of G and induces a star $K_{1,d}$ where $d = d_G(v) \geq 1$. Now, by induction, $\mu(G) = d \cdot \mu(G - N[v]) \leq d \cdot f(n - (d+1))$.

If $d \in \{1, 2\}$ or $d \geq 5$, then it is easily verified that $d \cdot f(n - (d+1)) < f(n)$ and hence $\mu(G) < f(n)$ in these cases.

If $d = 3$, then $d \cdot f(n-(d+1)) \leq f(n)$ with equality if and only if $n \equiv 1 \bmod 3$. Hence $\mu(G) \leq f(n)$. Furthermore, if $\mu(G) = f(n)$, then $\mu(G - N[v]) = f(n-(d+1))$ and hence, by induction, $G - N[v] = \frac{n-4}{3}K_3$. Now $G = K_{1,3} \cup \frac{n-4}{3}K_3 \in \mathcal{F}$.

If $d = 4$, then $d \cdot f(n-(d+1)) \leq f(n)$ with equality if and only if $n \equiv 2 \bmod 3$. Hence $\mu(G) \leq f(n)$. Furthermore, if $\mu(G) = f(n)$, then $\mu(G - N[v]) = f(n-(d+1))$ and hence, by induction, $G - N[v] = \frac{n-5}{3}K_3$. Now $G = K_{1,4} \cup \frac{n-5}{3}K_3 \in \mathcal{F}$. □

By Claims 1 to 4, the extremal graph G belongs to $\mathcal{F}$ or G is a $\{C_3, C_4\}$-free graph of minimum degree at least 2. Since $f(n) \geq 4 \cdot 3^{-\frac{5}{3}} \cdot 3^{\frac{n}{3}}$ and $0.928 \cdot \phi^{\frac{n}{2}} < 4 \cdot 3^{-\frac{5}{3}} \cdot 3^{\frac{n}{3}}$ for $n \geq 6$, Lemma 3.2 implies $\mu(G) < f(n)$, which completes the proof. □

3.4.2 Triangle-free graphs

Theorem 3.3. *If G is a triangle-free graph of order n, then $\mu(G) \leq g(n)$ where*

$$g(n) = \begin{cases} 1 & \text{, if } n = 1, \\ n-1 & \text{, if } n \in \{2,3,6,7\}, \\ 20 & \text{, if } n = 11, \text{ and} \\ 3^t \cdot 4^{\frac{n-4t}{5}} & \text{, if } n \geq 4t,\ n \equiv -t \bmod 5, \text{ for } t = 0,1,2,3 \text{ and } 4. \end{cases}$$

Furthermore, if the triangle-free graph G of order n with $n \geq 2$ is such that $\mu(G) = g(n)$, then $G \in \mathcal{G}$ where

$$\begin{aligned} \mathcal{G} &= \{K_{1,n-1} : 2 \leq n \leq 7\} \cup \{K_{1,2} \cup K_{1,3}, K_{1,4} \cup K_{1,5}\} \\ &\cup \left\{ tK_{1,3} \cup \frac{n-4t}{5}K_{1,4} : n \geq 4t, n \equiv -t \bmod 5,\ \text{for } t = 0,1,2,3 \text{ and } 4 \right\}. \end{aligned}$$

Proof of Theorem 3.3

Let G be a triangle-free graph of order n and size m. We prove, by induction on $n+m$, that

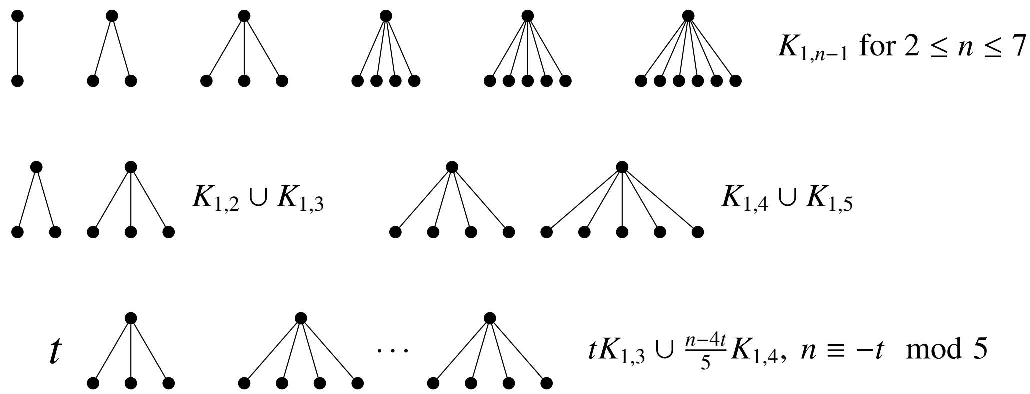

Figure 3.6: Family $\mathcal{G}$. Extremal graphs among triangle-free graphs.

$\mu(G) \leq g(n)$ and, for $n \geq 2$, $\mu(G) = g(n)$ if and only if G belongs to $\mathcal{G}$. Since the result is easily verified for $n \leq 8$, we assume now that $n \geq 9$. We establish a series of claims concerning properties that G can be assumed to have if G has maximum $\mu(G)$ among all graphs of order n.

Claim 5. *Every edge of G belongs to some DIM of G.*

Proof of Claim 5. This can be proved exactly as Claim 1. □

Claim 5 implies that G is $\{C_3, C_4\}$-free.

Claim 6. *The graph G has no isolated vertex.*

Proof of Claim 6. Note that unlike the function f from Theorem 3.2, the function g is strictly increasing for $n \geq 3$. Using this fact, this claim can be proved as Claim 3. □

Claim 7. *The graph G has minimum degree at least 2.*

Proof of Claim 7. By Claim 6, the graph G has no isolated vertex. Let u be a vertex of degree 1 and let v be the unique neighbor of u in G. Arguing as in the proof of Claim 4, we obtain that the closed neighborhood $N[v]$ of v in G is the vertex set of a component of G

and induces a star $K_{1,d}$ where $d = d_G(v) \geq 1$. Now, by induction, $\mu(G) = d \cdot \mu(G - N[v]) \leq d \cdot g(n - (d + 1))$.

It is easy to verify $d \cdot g(n - (d + 1)) \leq g(n)$ for every $n \geq 9$ with equality if and only if

- either $d = 3$, $n \not\equiv 0 \mod 5$ and $n \neq 11$,
- or $d = 4$ and $n \notin \{12, 16\}$,
- or $d = 5$ and $n = 11$.

The proof can now be completed similarly as the proof of Claim 4. We give details only for $d = 3$.

Let $d = 3$. We obtain $\mu(G) = d \cdot \mu(G - N[v]) \leq d \cdot g(n - (d + 1)) \leq g(n)$. If $\mu(G) = g(n)$, then $d \cdot g(n - (d + 1)) = g(n)$, which implies $n \bmod 5 \neq 0$ and $n \neq 11$. Furthermore, $\mu(G - N[v]) = g(n - (d + 1))$, which implies, by induction, that $G - N[v] \in \mathcal{G}$. Since for every graph H in $\mathcal{G}$ of order $n' = n - 4$ with $n' \geq 5$, $n' \not\equiv 1 \mod 5$, and $n' \neq 7$, we have $K_{1,3} \cup H \in \mathcal{G}$, we obtain $G \in \mathcal{G}$. □

By Claims 5 to 7, the extremal graph G belongs to $\mathcal{G}$ or G is a $\{C_3, C_4\}$-free graph of minimum degree at least 2. Clearly, $g(n) > 0.928 \cdot \phi^{\frac{n}{2}}$ for $n \in \{9, 10, 11\}$. Furthermore, for $n \geq 12$, we have $g(n) \geq 81 \cdot 4^{\frac{n-16}{5}} > 0.956 \cdot 1.319^n > 0.928 \cdot \phi^{\frac{n}{2}}$, Lemma 3.2 implies $\mu(G) < g(n)$, which completes the proof. □

3.4.3 Connected graphs

For an integer n with $n \geq 11$ and $n \equiv 1 \bmod 5$, let the graph H_n arise from $K_1 \cup \frac{n-1}{5} K_{1,4}$ by adding edges between K_1 and each center of the $\frac{n-1}{5}$ stars. See Figure 3.7. Let the graph H_8 of order 8 be as shown in Figure 3.8.

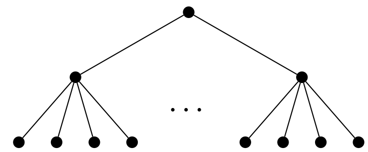

Figure 3.7: Graphs H_n for $n \equiv 1 \bmod 5$ and $n \geq 11$.

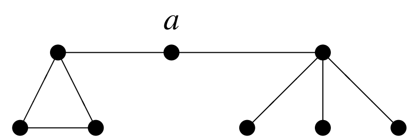

Figure 3.8: The graph H_8.

Theorem 3.4. *If G is a connected graph of order n, then $\mu(G) \leq h(n)$ where*

$$h(n) = \begin{cases} 1 & , \text{ if } n \in \{1, 2\}, \\ 3 & , \text{ if } n = 3, \\ n-1 & , \text{ if } 4 \leq n \leq 8, \text{ and} \\ 4^{\frac{n-1}{5}} & , \text{ if } n \geq 9. \end{cases}$$

Furthermore, if the connected graph G of order n is such that $\mu(G) = h(n)$, then $G \in \mathcal{H}$ where

$$\mathcal{H} = \{K_1, K_2, K_3, K_{1,3}, K_{1,4}, K_{1,5}, K_{1,6}, K_{1,7}, H_8\} \cup \{H_n : n \geq 11 \text{ and } n \equiv 1 \bmod 5\}.$$

Proof of Theorem 3.4

Let G be a connected graph of order n and size m. We prove the statement by induction on $n + m$. For $n \leq 8$, the result is easily verified.

Note that:

$$\text{for every } p \in \mathbb{N}, \text{ we have } p \cdot 4^{\frac{-(p+1)}{5}} \leq 1 \text{ with equality if and only if } p = 4. \tag{3.3}$$

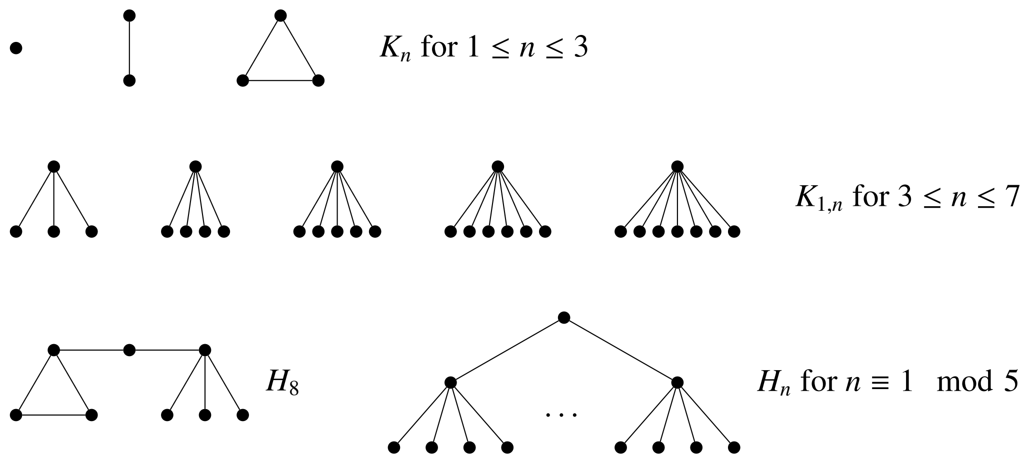

Figure 3.9: Family $\mathcal{H}$. Extremal graphs among connected graphs.

This implies that if $n \leq 8$ and G is neither a star nor a triangle nor H_8, then $n \geq 4$ and $\mu(G) \leq h(n) - 1 = n - 2 \leq 4^{\frac{n-1}{5}}$.

We assume now that $n \geq 9$. Note that if G' is a graph of order n' less than n and no component of G' is a star or a triangle or H_8, then, by induction, every component K of G of order $n(K)$ satisfies $\mu(K) \leq 4^{\frac{n(K)-1}{5}}$, which implies $\mu(G') \leq 4^{\frac{n'-1}{5}}$. We establish a series of claims concerning properties that G can be assumed to have if G has maximum $\mu(G)$ among all graphs of order n.

Claim 8. *Every edge of G that does not belong to some DIM of G is a bridge.*

Proof of Claim 8. If G contains an edge e such that no DIM of G contains e and e is not a bridge of G, then every DIM of G is a DIM of the connected graph $G - e$ and, by induction, $\mu(G) \leq \mu(G - e) \leq h(n)$. If $\mu(G) = h(n)$, then $\mu(G - e) = h(n)$ and hence, by induction, $G - e \in \mathcal{H}$. It is easily verified that adding any edge to a graph H in $\mathcal{H}$ results in a graph with strictly less DIMs than H. Therefore, $\mu(G) < \mu(G - e) = h(n)$, which is a contradiction. □

By Claim 8, the graph G has no cycle of length 4.

Claim 9. *No edge of G that does not belong to some DIM of G is incident with a vertex of degree* 1.

Proof of Claim 9. If uv is an edge of G that does not belong to some DIM of G such that u has degree 1, then $\mu(G) \leq \mu(G-u) \leq h(n-1) < h(n)$. □

Claim 10. *The graph G is triangle-free.*

Proof of Claim 10. Let $T = G[xyz]$ be a triangle in G. Since G is connected, we may assume that z has a neighbor z' that does not lie on T.

First, we assume that y has a neighbor y' that does not lie on T. Since G has no cycle of length 4, the vertices y' and z' are distinct. For every DIM M of G, the set M contains an edge of T and $M \setminus E(T)$ is a DIM of $G - V(T)$. This implies, by induction,

$$\begin{aligned}
\mu(G) &= \mu(G;\{x,y\},\emptyset) + \mu(G;\{x,z\},\emptyset) + \mu(G;\{y,z\},\emptyset) \\
&\overset{(Lemma\ 3.1)}{=} \mu(G;\{x,y,z'\},\{y'\}) + \mu(G;\{x,z,y'\},\{z'\}) + \mu(G;\{y,z\},\{y',z'\}) \\
&\leq \mu(G-V(T);\{z'\},\{y'\}) + \mu(G-V(T);\{y'\},\{z'\}) + \mu(G-V(T);\emptyset,\{y',z'\}) \\
&\leq \mu(G-V(T)) \\
&\leq h(n-3) \\
&< h(n).
\end{aligned}$$

Hence we may assume that for every triangle $\tilde{T}$ of G, exactly one vertex of $\tilde{T}$ has degree at least 3.

Next, we assume that no component of $G - V(T)$ is either a star or a triangle or H_8. By

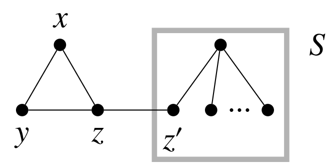

Figure 3.10: Graph determined by a triangle, deg(z) = 3 and a star S.

induction, this implies that $\mu(G - V(T)) \leq 4^{\frac{(n-3)-1}{5}}$. Now

$$\begin{aligned} \mu(G) &= \mu(G; \{x, y\}, \emptyset) + \mu(G; \{x, z\}, \emptyset) + \mu(G; \{y, z\}, \emptyset) \\ &\overset{(3.1)}{=} \mu(G; \{x, y, z'\}, \emptyset) + \mu(G; \{x, z\}, \{z'\}) + \mu(G; \{y, z\}, \{z'\}) \\ &\leq \mu(G - V(T); \{z'\}, \emptyset) + 2 \cdot \mu(G - V(T); \emptyset, \{z'\}) \\ &\leq 2 \cdot \mu(G - V(T)) \\ &\leq 2 \cdot 4^{\frac{(n-3)-1}{5}} \\ &< 4^{\frac{n-1}{5}}. \end{aligned}$$

Hence we may assume that for every triangle $\tilde{T}$ of G, some component of $G - V(\tilde{T})$ is either a star or a triangle or H_8.

Next, we assume that some component S of $G - V(T)$ is a star of order s. Since the edge xy belongs to some DIM of G, we obtain that $s \geq 2$. Since the edge xz belongs to some DIM of G, we obtain that $s \geq 3$, that z is adjacent to a leaf z' of S, and z is not adjacent to another vertex in S. If z has degree 3, then the graph is completely determined. See Figure 3.10 Note that the structure of G is similar to H_8 in this case. Using $n \geq 9$, it is easy to verify that $\mu(G) < h(n)$. Hence we may assume that z has degree at least 4. If $G - V(S)$ is H_8, then the graph is completely determined. See Figure 3.11 Again it is easy to verify that $\mu(G) < h(n)$. Hence no component of $G - V(S)$ is a either a star or a triangle or H_8, which implies, by induction, $\mu(G - V(S)) \leq 4^{\frac{(n-s)-1}{5}}$. Since every DIM of G contains

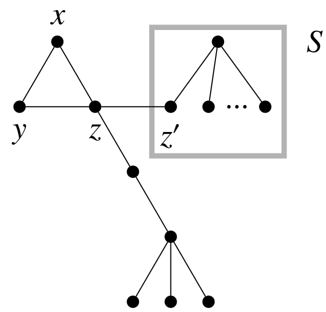

Figure 3.11: Graph determined by H_8, a star S and $\deg(z) = 4$.

an edge of S, we obtain, by induction,

$$\begin{aligned}
\mu(G) &= \mu(G; \{z'\}, \emptyset) + \mu(G; \emptyset, \{z'\}) \\
&\overset{(3.1)}{=} \mu(G; \{z'\}, \{z\}) + \mu(G; \{z\}, \{z'\}) \\
&\leq \mu(G - V(S); \emptyset, \{z\}) + (s-2) \cdot \mu(G - V(S); \{z\}, \emptyset) \\
&\leq (s-2) \cdot \mu(G - V(S)) \\
&\leq (s-2) \cdot 4^{\frac{(n-s)-1}{5}} \\
&\overset{(3.3)}{<} 4^{\frac{n-1}{5}}.
\end{aligned}$$

Hence we may assume that for every triangle $\tilde{T}$ of G, no component of $G - V(\tilde{T})$ is a star.

Next, we assume that some component T' of $G - V(T)$ is a triangle. Since $n \geq 9$, the degree of z is at least 4. This implies that the connected graph $G - V(T')$ is H_8. Now the graph is completely determined, $n = 11$, and $\mu(G) = 8 < h(n)$ (see Figure 3.12 (i)). Hence we may assume that for every triangle $\tilde{T}$ of G, some component of $G - V(\tilde{T})$ is H_8. Now the graph G arises from $T \cup H_8$ by adding an edge between z and the vertex a in H_8 (see Figure 3.12 (ii)). This implies $n = 11$ and $\mu(G) = 13 < h(n)$, which completes the proof of the claim. □

Claims 8 and 10 imply that G is $\{C_3, C_4\}$-free. By assumption, G has no isolated vertex.

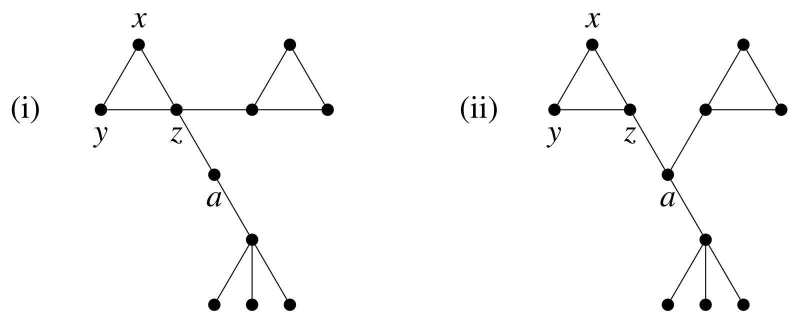

Figure 3.12: Graphs determined by H_8 and a triangle.

Claim 11. *The graph G has minimum degree at least* 2*.*

Proof of Claim 11. Let v be a vertex of G of degree $p+q$ such that v has $p \geq 1$ neighbors $u_1, \ldots, u_p$ of degree 1 and q neighbors $w_1, \ldots, w_q$ of degree at least 2. If G is a star, then the theorem is easily verified. Hence we may assume that $q \geq 1$. Since every DIM of G contains an edge incident with v, every edge between a vertex in $N[v]$ and $V(G) \setminus N[v]$ is a bridge. Since G is triangle-free, this implies that every edge incident with a vertex in $N[v]$ is a bridge and that $N[v]$ induces a star S. For $1 \leq j \leq q$, let z_j denote a neighbor of w_j that is distinct from v. Let $Z = \{z_1, \ldots, z_q\}$.

We have

$$\begin{aligned}
\mu(G) &= \sum_{i=1}^{p} \mu(G; \{v, u_i\}, \emptyset) + \sum_{j=1}^{q} \mu(G; \{v, w_j\}, \emptyset) \\
&\overset{(3.1)}{=} \sum_{i=1}^{p} \mu(G; \{v, u_i\} \cup Z, \emptyset) + \sum_{j=1}^{q} \mu(G; \{v, w_j\} \cup (Z \setminus \{z_j\}), \{z_j\}) \\
&\leq \sum_{i=1}^{p} \mu(G - V(S); Z, \emptyset) + \sum_{j=1}^{q} \mu(G - V(S); Z \setminus \{z_j\}, \{z_j\}) \\
&= p \cdot \mu(G - V(S); Z, \emptyset) + \sum_{j=1}^{q} \mu(G - V(S); Z \setminus \{z_j\}, \{z_j\}) \\
&\leq p \cdot \mu(G - V(S)).
\end{aligned}$$

If $q \geq 2$ and some component S' of $G - V(S)$ is a star, then Claim 9 implies that S' has order at least 2 and, by exchanging the roles of S and S', we may assume that $q = 1$. Hence we may assume that

- either $q \geq 2$ and no component of $G - V(S)$ is a star
- or $q = 1$.

If $q \geq 2$ and no component of $G - V(S)$ is a star, then, by induction, $\mu(G - V(S)) \leq 4^{\frac{(n-|V(S)|)-1}{5}} = 4^{\frac{(n-(p+q+1))-1}{5}} \leq 4^{\frac{(n-(p+3))-1}{5}}$ and we obtain $\mu(G) \leq p \cdot \mu(G - V(S)) \leq p \cdot 4^{\frac{(n-(p+3))-1}{5}} \overset{(3.3)}{<} 4^{\frac{n-1}{5}}$. Hence we may assume now that $q = 1$.

First, we assume that the edge vw_1 does not belong to any DIM of G. In this case $\mu(G) \leq p \cdot \mu(G - \{u_1, \ldots, u_p, v\})$. If the connected graph $G - \{u_1, \ldots, u_p, v\}$ is a star, then the result is easily verified. Hence we may assume that $G - \{u_1, \ldots, u_p, v\}$ is not a star. By induction, this implies $\mu(G - \{u_1, \ldots, u_p, v\}) \leq 4^{\frac{(n-(p+1))-1}{5}}$ and hence $\mu(G) \leq p \cdot \mu(G - \{u_1, \ldots, u_p, v\}) \leq p \cdot 4^{\frac{(n-(p+1))-1}{5}} \overset{(3.3)}{\leq} 4^{\frac{n-1}{5}}$. Furthermore, if $\mu(G) = 4^{\frac{n-1}{5}}$, then, by (3.3), we have $p = 4$, $n \equiv 1 \mod 5$, and $\mu(G - \{u_1, \ldots, u_p, v\}) = 4^{\frac{(n-(p+1))-1}{5}}$. By induction, this implies $G - \{u_1, \ldots, u_p, v\} = H_{n-5}$, which easily implies $G = H_n \in \mathcal{H}$. Hence we may assume that the edge vw_1 belongs to some DIM of G.

Next, we assume that some component S' of $G - V(S)$ is a star of order s'. Since, by Claim 9, the edge u_1v belongs to some DIM of G, we obtain that $s' \geq 2$. Since the edge vw_1 belongs to some DIM of G, we obtain that $s' \geq 3$ and that w_1 is adjacent to a leaf z' of S'. If w_1 has degree 2, then the graph is completely determined (see Figure 3.13) and it is easy to verify that $\mu(G) < h(n)$. Hence we may assume that w_1 has degree at least 3. Since vw_1 belongs to some DIM of G, the graph $G - V(S')$ does not belong to $\mathcal{H}$ and $n - s' \geq 6$. By induction, this implies $\mu(G - V(S')) \leq 4^{\frac{(n-s')-1}{5}}$. Since every DIM of G contains an edge

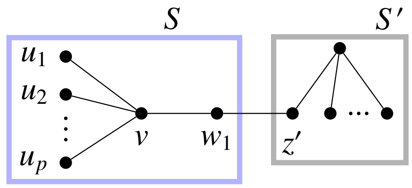

Figure 3.13: Graph determined by stars S and S'.

of S', we obtain

$$\begin{aligned}
\mu(G) &= \mu(G;\{z'\},\emptyset) + \mu(G;\emptyset,\{z'\}) \\
&\overset{(3.1)}{=} \mu(G;\{z'\},\{w_1\}) + \mu(G;\{w_1\},\{z'\}) \\
&\leq \mu(G - V(S');\emptyset,\{w_1\}) + (s'-2)\cdot\mu(G - V(S');\{w_1\},\emptyset) \\
&\leq (s'-2)\cdot\mu(G - V(S')) \\
&\leq (s'-2)\cdot 4^{\frac{(n-s')-1}{5}} \\
&\overset{(3.3)}{<} 4^{\frac{n-1}{5}}.
\end{aligned}$$

Hence we may assume that no component of $G - V(S)$ is a star. By induction, this implies $\mu(G - V(S)) \leq 4^{\frac{(n-(p+2))-1}{5}}$ and we obtain $\mu(G) \leq p\cdot\mu(G - V(S)) \leq p\cdot 4^{\frac{(n-(p+2))-1}{5}} \overset{(3.3)}{<} 4^{\frac{n-1}{5}}$, which completes the proof of the claim. □

By Claims 8 to 11, if the graph G is extremal and $G \notin \mathcal{H}$, G is a $\{C_3, C_4\}$-free graph of minimum degree at least 2. Since $0.928\cdot\phi^{\frac{n}{2}} < 4^{\frac{n-1}{5}}$ for $n \geq 9$, Lemma 3.2 implies $\mu(G) < h(n)$, which completes the proof. □

Resumen del Capítulo 3

Los problemas que combinan propiedades de cubrimiento y empaquetamiento en grafos han despertado interés recientemente. Entre ellos se encuentran el problema del conjunto dominante eficiente de aristas (también conocido como matching inducido dominante o DIM por sus siglas en inglés) y el problema del conjunto perfectamente dominante de aristas (que será tratado en el próximo capítulo).

En este capítulo damos un enfoque combinatorio al problema de dominación eficiente de aristas. Damos cotas ajustadas para el máximo número posible de DIMs para un grafo que es arbitrario, sin triángulos o conexo. Más aún, caracterizamos todos los grafos extremales para estas clases. Nuestros resultados implican que si G es un grafo de orden n y $\mu(G)$ es la cantidad de DIMs en el grafo, entonces $\mu(G) \leq 3^{\frac{n}{3}}$; $\mu(G) \leq 4^{\frac{n}{5}}$ cuando G no tiene triángulos; y $\mu(G) \leq 4^{\frac{n-1}{5}}$ cuando G es conexo y $n \geq 9$.

En la Sección 3.1 repasamos las definiciones necesarias.

Una arista *domina* a sí misma y a toda arista que comparta un vértice con ella. Un subconjunto de aristas $E' \subset E(G)$ es *dominante* si toda arista de G está dominada por al menos una arista en E'. Además $E' \subset E(G)$ es un *conjunto eficientemente dominante de aristas* si toda arista de G es dominada por exactamente una arista en E'. Un *matching inducido* es un conjunto de aristas $M \subset E(G)$ tal que toda arista en G es adyacente a lo sumo a una arista en M. Un *matching inducido dominante* (o DIM por sus siglas en ingés) es un matching inducido que además es dominante. En consecuencia, un conjunto eficientemente dominante de aristas es equivalente a un matching inducido dominante.

No todo grafo tiene un DIM; por ejemplo C_4 o cualquier ciclo C_n con $n \neq 3k$ no admiten DIMs. El problema de decidir si un grafo admite uno es conocido en la literatura como el problema de la dominación eficiente de aristas o problema DIM; y es NP-completo [43]. Denotamos con $\mu(G)$ el número de matching inducidos dominantes del grafo G. El problema de hallar $\mu(G)$ es la versión de conteo del problema DIM.

No es difícil de ver que un DIM es también un matching inducido máximo [17], y por lo tanto, todos los DIMs tienen el mismo tamaño.

Los grafos que consideramos en este capítulo no son necesariamente conexos. Claramente un DIM en un grafo es la unión de un DIM de cada una de sus compenentes conexas. Entonces, la cantidad de DIMs en un grafo no conexo es el producto del número de DIMs en cada una de sus componentes conexas.

In Section 3.3 usamos una definición alternativa del problema DIM. Ésta consiste en determinar si existe una partición de los vértices en dos subconjuntos disjuntos; de manera que: un subconjunto, los vértices blancos W, sea un conjunto independiente, mientras que el otro subconjunto, los vértices negros B, sea un grafo 1-regular.

Usando estos conceptos obtenemos algunos resultados auxiliares entre los que se destaca la cota del Lema 3.2 para la cantidad de DIMs en un grafo sin C_3 ni C_4 y mínimo grado por lo menos 2.

En la Sección 3.4 enunciamos y probamos las cotas ajustadas para $\mu(G)$ y caracterizamos los grafos extremales para la clase general de grafos (Teorema 3.2), grafos sin triángulos (Teorema 3.3) y grafos conexos (Teorema 3.4).

Chapter 4

Perfect Edge Domination

Edge dominating set problems have been the focus of considerable attention, in the last few years. Among the relevant variations of these problems are the perfect edge domination and efficient edge domination. In the former variation, each edge not in the dominating set is dominated by exactly one edge, while in the latter, every edge of the graph is so dominated. (See Figures 4.1 and 4.2.) These two types of dominations may lead to problems of a quite different nature. To start with, an efficient edge dominating set of a graph may not exist, while it necessarily exists for the perfect domination (for example, take all edges of the graph). Furthermore, there are important differences concerning the complexity status of the problems for classes of graphs although both of them are hard, in general. In fact, perfect edge dominating set problems seem to be not easier to tackle than efficient edge dominating set problems. On the other hand, if a graph contains an efficient edge dominating set then such set is also a minimum cardinality perfect edge dominating set [41, 63].

There are many results concerning the efficient edge dominating case as we mention in Chapter 3. Some of them are helpful to approach perfect edge dominating set problems.

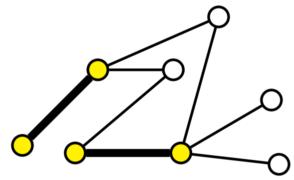

Figure 4.1: An efficient edge dominating set.

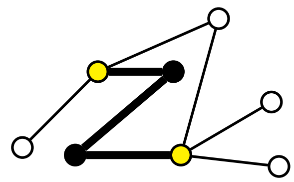

Figure 4.2: A perfect edge dominating set.

Known results for the perfect edge dominating set problem are as follows. Lu, Ko and Tang [63] proved the problem is NP-complete for bipartite graphs. As for polynomial time solvable cases, there are linear time algorithms also described by Lu, Ko and Tang, for generalized series-parallel graphs and chordal graphs. In addition, there is a linear time algorithm for circular-arc graphs, by Lin, Mizrahi and Szwarcfiter [60]. All of these algorithms solve the weighted perfect edge dominating set problem.

We point out that our contributions of the chapter form part of Lin et al. [56]. Many other results of the latest work are also described in this present chapter. This chapter is organized as follows. In Section 4.1 we recall some definitions and introduce a 3-coloring of the vertices to classify different types of perfect edge dominating sets. In Section 4.2 we study the complexity of the problem and prove that the weighted version is linear-time solvable for cubic claw-free graphs (degree exactly three). Note that the problem is NP-complete for subcubic claw-free graphs (degree at most three) [56]. In Section 4.3 we use the 3-coloring of the vertices to state properties of P_k-free graphs. We also analyze the relationship between dominating subgraphs and different types of perfect edge dominating sets and present a robust linear-time algorithm to solve the weighted version for P_5-free graphs.

4.1 3-Colorings associated to a edge dominating set

A subset of edges $E' \subseteq E(G)$ is a *perfect edge dominating set* of G, if every edge of $E(G) \setminus E'$ is dominated by exactly one edge of E'. On the other hand if every edge of $E(G)$ is dominated exactly once by E' then E' is an *efficient edge dominating set*, also called a *dominating induced matching*.

The *cardinality perfect (efficient) edge dominating set problem* is to determine the perfect (efficient) edge dominating set of G of least cardinality. The corresponding *weighted* problems are defined replacing minimum cardinality by minimum sum of weights of the dominating edges.

3-Coloring

Let $P \subseteq E(G)$ be a perfect edge dominating set of a connected graph G. Then P defines a 3-coloring of the vertices of G, as below:

- *black vertices*, B: those having at least two incident edges of P;
- *yellow vertices*, Y: those which are incident to exactly one edge of P;
- *white vertices*, W: the ones not incident to any edge of P.

We call (B, Y, W) the *3-coloring associated to P*.

Observation 4.1. *W is an independent set.*

Observation 4.2. *Pendant vertices (vertices of degree 1) in $G \setminus W$ are exactly the yellow vertices.*

Observation 4.3. *White vertices have only yellow neighbors.*

Observation 4.4. *Every vertex of an induced K_t with $t \geq 4$ must be black.*

Observation 4.5. *Every induced triangle has (i) three black vertices or (ii) two yellow vertices and one white vertex.*

It is straightforward to see that any 3-coloring (B, Y, W) of the vertices of G, that verifies the conditions of Observations 4.1, 4.2 and 4.3, is necessarily associated to some perfect edge dominating set of G.

On the other hand, an efficient edge dominating set is clearly a perfect edge dominating set. In this case, the set B is empty. The latter implies that the subset induced by Y is an induced matching of G.

When $B \neq \emptyset$, P is not an efficient edge dominating set. In particular, $E(G)$ is a perfect edge dominating set called the *trivial perfect edge dominating set*. In this case, $W = \emptyset$.

Any perfect edge dominating set P which is neither trivial nor an efficient edge dominating set is called *proper perfect edge dominating set* because $W, B, Y \neq \emptyset$. Similarly, (B, Y, W) is then called a *proper coloring*.

Along this chapter, we consider these three types of perfect edge dominating sets:

- efficient, when the 3-coloring associated satisfies $B = \emptyset$;
- trivial, when the 3-coloring associated satisfies $W = \emptyset$;
- proper, when the 3-coloring associated satisfies $B \neq \emptyset$, $Y \neq \emptyset$, and $W \neq \emptyset$.

4.2 Complexity for H-free graphs

Since the perfect edge dominating set problem is NP-hard in general, in this section we states some properties that allows us to determine the complexity of the problem in several graph classes. Lin et al. [56] gave an interesting analysis about the complexity of this problem. We mention below some results from this manuscript with the propose to have the correct context and motivate the interest to consider these graph classes.

Theorem 4.1. *[56] The minimum perfect edge dominating set problem for claw-free graphs of degree at most 3 is NP-hard.*

We will show that for cubic graphs (degree exactly three) the problem becomes linear. In consequence, it is shown that when the problem is restricted to claw-free graphs, the complexity strongly depends on the degree of the vertices.

Define a *linear forest* as a graph whose connected components are induced paths.

The following theorem determine the complexity of the problem on graph classes defined by forbidding a particular subgraph and a restriction on the maximum degree.

Theorem 4.2. *[56] Let H be a graph, and $\mathcal{G}$ the class of H-free graphs of degree at most d, for some fixed $d \geq 3$. Then the perfect edge dominating set problem is*

- *polynomial time solvable for graphs in $\mathcal{G}$ if H is a linear forest*
- *NP-complete otherwise.*

Corollary 4.1. *[56] The perfect edge dominating set problem is NP-complete for H-free graphs where H is any graph except linear forests.*

In last section of this chapter we will analyze the case $H = P_k$.

4.2.1 Cubic claw-free graphs

In this section, we show that the weighted version of the perfect edge dominating set problem can be solved in linear time for cubic claw-free graphs. Without loss of generality, we consider only connected graphs.

We use the following result of [59] concerning efficient edge domination in claw-free graphs.

Lemma 4.1. *[59] There is an algorithm of complexity $O(n)$ which finds the least weight efficient edge dominating set of a claw-free graph G, or reports that G does not admit efficient edge dominating sets.*

Next properties arise when we additionally restrict the degree of the vertices in a claw-free graph to be exactly 3 (cubic graph).

Observation 4.6. *A graph G is a cubic claw-free graph if and only if G is cubic and every vertex of it is contained in some triangle.*

Lemma 4.2. *Let G be a connected cubic claw-free graph. Then G admits no proper perfect edge dominating set.*

Proof. Suppose the contrary and let P be a proper perfect edge dominating set of G, and (B,Y,W) its corresponding coloring of the vertices of G. The idea is to show that every vertex of G has black color which implies that P is the trivial perfect edge dominating set, which is a contradiction.

Since P is proper, G contains some black vertex v. By Observation 4.6, v is contained in at least one triangle of G. Clearly, any pair of adjacent neighbors of v must have black color by Observation 4.5. We can apply iteratively the same reasoning to each new considered black vertex and we call this procedure as *black propagation*.

If every vertex of G has been considered, we are done. Suppose there is some unconsidered vertex. As G is connected, there must be an edge uw in G such that u has black color and w not. It is clear that u and w do not have a common neighbor. By Observation 4.6 there is a triangle C' containing w. Since u has black color, by Observation 4.3, w cannot be a white vertex. By Observation 4.5, w must have some neighbor with non-white color in C'. But w has already a black neighbor u which is not in C'. Therefore, w has black color by Observation 4.2. Again, we can apply the black propagation to w.

Repeating iteratively the same argument, we conclude that all vertices of G must be black, and therefore G cannot contain a proper edge dominating set. □

The algorithm follows immediately from the two above lemmas.

Algorithm for minimum weighted perfect edge dominating set

Let G be a connected cubic claw-free graph, whose edges have been assigned weights, possibly negative. By Lemma 4.2 it does not contain proper perfect edge dominating sets. Then first, we apply the algorithm described in [59] for claw-free graphs. If G contains an efficient edge dominating set then the latter algorithm finds the least weighted of such sets, and the minimum between this set and the trivial perfect edge dominating set $E(G)$ is the answer of our algorithm. Otherwise, G does not contain an efficient edge dominating set, and therefore the only perfect edge dominating set of G is $E(G)$ and this is the answer of the algorithm.

By recalling that $|E(G)| = \frac{3n}{2}$, the complexity of the algorithm is therefore $O(n)$. Thus, we have proved the following theorem.

Theorem 4.3. *There is a linear-time algorithm that solves the weighted perfect edge dominating set problem in cubic claw-free graphs.*

This linear-time algorithm can be extended easily to claw-free graphs of degree at most three where every degree two vertex has two adjacent neighbors.

4.3 Graphs without paths of length 5

Motivated by the Corollary 4.1 [56], it is an open problem to determine the complexity of perfect dominating set problem for H-free graphs with $H = P_k$, (any line forest F has some induced subgraph P_k and F is induced subgraph of some $P_{k'}$, $k \leq k'$). In this sense, we have done the work for $k = 5$. Our main objective is a robust and efficient approach to the weighted problem. Despite of it, the observations and subroutines described below have independent interest and can be used for other applications.

We first present structural properties of P_5 graph, and describe the theorems in which the correctness and complexity of the proposed algorithm is based on.

4.3.1 Properties of P_5-free graphs

The following definitions are useful. The *eccentricity* of a vertex v, denoted $\epsilon(v)$ is the maximum distance between v and any other vertex. A vertex of minimum eccentricity in the graph is called *central*. Finally, a vertex having eccentricity at most 2 is named a *principal vertex*.

Note that if a connected graph has a vertex with eccentricity at least 4 then it has an induced P_5. Any central vertex of a connected P_5-free graph has eccentricity at most 2 (consequence of Theorem 4.4).

The following structural result by Bacsó and Tuza is fundamental for our approach.

Theorem 4.4. *[2] Every connected graph contains an induced P_5, a dominating K_p, or a dominating P_3.*

The proposed algorithm needs to determine which of these three subgraphs G contains. This will be achieved through a principal vertex. Then we need a robust method to compute such a vertex, if it exists. For a chosen vertex v, call *Test(v)* the operation that determines its eccentricity and recognizes the following situations:

(i) $\epsilon(v) = \infty$, and then G is not connected;

(ii) $\epsilon(v) \geq 4$, and then G has an induced P_5;

(iii) $\epsilon(v) = 3$, and then G has an induced P_4 starting in v;

(iv) $\epsilon(v) \leq 2$, and then v is a principal vertex of G.

It is clear that Test(v) can be done in linear-time. The cases (i) and (ii) allow to recognize that the input G is not a connected P_5-free graph.

The next theorem describes the robust linear time computation of a principal vertex.

Theorem 4.5. *Any connected graph G contains a principal vertex or an induced P_5. Moreover there is a linear-time algorithm to find a principal vertex of G or to detect that G is not P_5-free.*

Proof. We describe an algorithm that uses at most 3 times the procedure *Test* a vertex. At any point of the algorithm, if an induced P_5 is found, then the algorithm ends and returns that induced P_5. First choose any vertex $u \in V$ and Test(u). If $\epsilon(u) = 3$, let z be such that $dist(u, z) = 3$ and u, v, w, z an induced P_4. Choose $v_1 \in A = N(u) \cap N(w)$ such that v_1 has the maximum number of neighbors in $B = V \setminus (N(u) \cup N(w) \cup N(z))$. Note that u, v_1, w, z is an induced P_4 and possibly $v = v_1$.

Next, Test(v_1) and Test(w). We will show that $\epsilon(v_1) = \epsilon(w) = 3$ implies that G has an induced P_5. As v_1 has eccentricity 3, there exists a vertex x such that $dist(x, v_1) = 3$. Also x verifies $dist(x, w) = 3$ or $dist(x, w) = 2$ because $v_1 w \in E$.

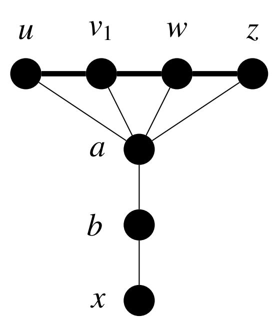

Figure 4.3: $dist(v_1, x) = dist(w, x) = 3$.

If $dist(x, w) = 3$ let v_1, a, b, x be a shortest path between v_1 and x, as in Figure 4.3. Note that $N[x] \cap \{u, v_1, w, z\} = \emptyset$, $N(b) \cap \{v_1, w\} = \emptyset$ because $dist(x, v_1) = 3$ and $dist(x, w) = 3$. If

$N(b) \cap \{u, z\} \neq \emptyset$ there is an induced P_5 and we can find it and return it. Otherwise, $N(b) \cap \{u, z\} = \emptyset$. We can assume $wa \in E$ otherwise w, v_1, a, b, x is an induced P_5. Following the same idea, x, b, a, v_1, u and x, b, a, w, z are possible induced P_5's, then we can assume that $ua, az \in E$ which is a contradiction because $dist(u, z) = 3$. Therefore, in this case the algorithm always finds an induced P_5.

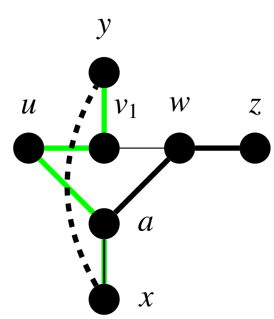

Figure 4.4: $dist(v_1, x) = 3$ and $dist(w, x) = 2$.

If $dist(x, w) = 2$ let w, a, x be a shortest path between w and x. Note that $N[x] \cap \{u, v_1, w\} = \emptyset$ because $dist(x, v_1) = 3$. If $zx \in E$ then u, v_1, w, z, x is an induced P_5. Thus, $zx \notin E$ and we are in the situation of Figure 4.4. We assume $ua \in E$ otherwise u, v_1, w, a, x is an induced P_5. This implies that $a \in A = N(u) \cap N(w)$ and it is adjacent to $x \in B = V \setminus (N(u) \cup N(w) \cup N(z))$. Recall that v_1 was chosen as a vertex belonging to A with maximum degree in B. As $v_1x \notin E$, then there exists $y \in B$ such that $v_1y \in E$ and $ay \notin E$. It follows that there is a P_5 induced either by y, v_1, u, a, x if $xy \notin E$ or by x, y, v_1, w, z if $xy \in E$. Again, the algorithm always finds an induced P_5. □

Once a principal vertex is obtained, we find an induced P_5, or a dominating K_p, or a dominating P_3 following the theorem below. We remark that using a recent characterization of P_k-free graphs, by Camby and Schaudt [16], it is possible to obtain a dominating induced P_3 or a dominating K_p in $O(n^5(n + m))$ time when a connected P_5-free graph is given.

Theorem 4.6. *For any connected graph G, there is a linear-time algorithm to find a dominating induced P_3, a dominating K_p, or to detect that G is not a P_5-free graph.*

Proof. The algorithm searches a dominating set contained in $N[v]$ where v is a principal vertex of G. Again, at any point of the algorithm, if an induced P_5 is found, then the algorithm ends and returns that induced P_5.

1. Find a principal vertex v or an induced P_5 of G using the robust linear-time algorithm of Theorem 4.5.
2. Let $X := N[v]$ be the initial dominating set. Consider iteratively each vertex $w \in N(v)$: if $X \setminus \{w\}$ is still a dominating set then $X := X \setminus \{w\}$. This can be done in linear-time using a variable for each vertex of $V \setminus N[v]$ to count the number of neighbors in X. A vertex $w \in N(v)$ can not be removed from X if only if some of its neighbors has exactly one dominator in X.

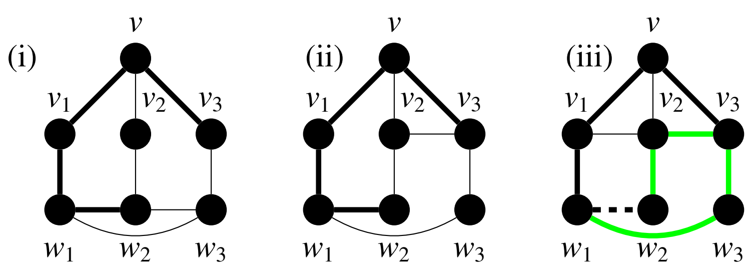

Figure 4.5: Principal vertex v, its neighbors v_1, v_2, v_3 and their proper dominated vertices.

3. If $|X| \le 3$ then G has a dominating P_3 or a dominating $K_{|X|}$. If $|X| \ge 4$ and it is not a complete graph, we will show that G has an induced P_5. We can assume that G has a subgraph like one in Figure 4.5, where $\{v_1, v_2, v_3\} \subset X$ and does not induce a triangle. Each one of them has a proper dominated vertex w_1, w_2, w_3 respectively. Edges between w_i and w_j are drawn in order to avoid the induced P_5 w_i, v_i, v, v_j, w_j. Nevertheless in (i) and (ii) w_2, w_1, v_1, v, w_3 is a induced P_5; and in (iii) the vertices w_2, v_2, v_3, w_3, w_1 or w_2, w_1, v_1, v, w_3 induce a P_5. □

We remark that the above theorem might be of interest to other P_5-free algorithmic problems, since it represents an algorithmic proof of Bacsó and Tuza's theorem [2].

4.3.2 Colorings and vertex dominating subgraphs

In this section, we describe the remaining algorithms on which the proposed algorithm is based. The first two theorems show the convenience of using 3-colorings to compute perfect edge dominating sets.

Theorem 4.7. *Given a graph G and $W \subseteq V$, there is a linear-time algorithm to verify if there exists some perfect edge dominating set whose associated 3-coloring is (B, Y, W).*

Proof. In the affirmative case, using Observation 4.2, we can determine Y in linear-time and so $B = V \setminus (W \cup Y)$. Therefore, we construct a 3-coloring (B, Y, W) in this way and then test the validity of (B, Y, W) by checking the conditions of Observations 4.1, 4.2 and 4.3. All these computations can be done in linear-time. □

Theorem 4.8. *Given a connected graph G and $Y \subset V$, there is a linear-time algorithm to verify if there exists some perfect edge dominating set whose associated 3-coloring is $(B = \{b\}, Y, W)$. Moreover, the algorithm can find a vertex b for which the sum of the weight of its incident edges is minimum.*

Proof. Clearly, if such perfect edge dominating set exists, it verifies

i. $G[Y]$ is a graph with maximum degree 1 and at least one vertex has degree 0.

ii. $V \setminus Y$ is a independent set.

iii. If $\{s_1, s_2, ..., s_k\} \subseteq Y$ are the vertices with degree 0 in $G[Y]$, and $\{y_1, y_2, ..., y_l\} \subset Y$ are the vertices of degree 1, the black vertex belongs to $\bigcap_{i=1}^{k} N(s_i) \setminus \bigcup_{j=1}^{l} N(y_j)$.

Therefore, we check conditions (i) and (ii), and construct $A = \bigcap_{i=1}^{k} N(s_i) \setminus \bigcup_{j=1}^{l} N(y_j)$ in linear-time. Note that all vertices in A are equivalent. If $A \neq \emptyset$, each $b \in A$ generates a valid coloring (B, Y, W), $B = \{b\}$, $W = V \setminus (B \cup Y)$. Condition (ii) and $|B| = 1$ imply that (B, Y, W) satisfies the conditions of Observations 4.1 and 4.3. Condition (i) and $b \in A$ imply that (B, Y, W) satisfies the condition of Observation 4.2. So, the validity of (B, Y, W) holds. □

The next theorem refers to finding an efficient edge dominating set of a graph, given a fixed size vertex dominating set of it.

Theorem 4.9. *[58] Given a graph G and a vertex dominating set of fixed size of G, there is a linear-time algorithm to solve the minimum weight efficient edge dominating set problem for G.*

The following results relate trivial perfect edge dominating sets of a graph and the existence of vertex dominating complete subgraphs of certain sizes.

Observation 4.7. *Given a graph G, a dominating set D of G, a perfect edge dominating set P of G and (B, Y, W) the 3-coloring associated to P, if $D \subseteq B$ then P is the trivial perfect edge dominating set.*

Corollary 4.2. *Given a connected graph G, if there is a dominating K_p with $p \geq 4$ then G has exactly one perfect edge dominating set P and P is trivial.*

Proof. Suppose that there is a non-trivial perfect edge dominating set P. By Observation 4.4, all vertices of K_p must be black and by Observation 4.7, P must be trivial and this is a contradiction. □

Corollary 4.3. *Given a connected P_5-free graph G, if G admits some non-trivial perfect edge dominating set P then G has dominating induced P_3 or a dominating K_3.*

Proof. If G has at most 2 vertices, then G has only one perfect edge dominating set which is trivial. Therefore, G has at least 3 vertices. Now, suppose that G has neither dominating induced P_3 nor dominating K_3. By Theorem 4.4, G must have some dominating K_p with $p \neq 3$. In case $p \geq 4$, by Corollary 4.2, G has no non-trivial perfect edge dominating set which is a contradiction. Therefore, $p \leq 2$. As G is connected with at least 3 vertices, it is always possible to add more vertices to K_p to form a dominating K_3 or a dominating induced P_3. Again, a contradiction. □

Corollary 4.4. *Given a graph G, if G has some vertex dominating K_1 or K_3 then G has no proper edge dominating sets.*

Proof. First, assume G has a dominating $K_1 = \{u\}$ which means that u is a universal vertex. Suppose that P is a proper edge dominating set. Let v be any black vertex. By Observation 4.7, v cannot be a universal vertex. As u is a neighbor of v, then u is a yellow vertex because it is a universal vertex. As v is a black vertex, v has another non-white neighbor w. Moreover, w is adjacent to the universal vertex u. This contradicts that u is a yellow vertex.

Next, suppose G contains a dominating K_3. Suppose that G has some proper edge dominating set P and (B, Y, W), its associated 3-coloring, with $B, Y, W \neq \emptyset$. By Observation 4.5, (i) all vertices of the dominating K_3 are black or (ii) exactly two of them are yellow vertices and the other vertex is white. In case (i), by Observation 4.7, P must be trivial which is a contradiction. In case (ii), there are no black vertices as consequence of Observations 4.2 and 4.3. Again, a contradiction. □

Finally, we state the relationship between the existence of proper edge dominating sets and the colorings of a dominating P_3.

Theorem 4.10. *Given a connected P_5-free graph G, if G admits some proper perfect edge dominating set P, with associated 3-coloring (B, Y, W), then G has a vertex dominating P_3 (formed by vertices v_1, v_2 and v_3) with one of the possible combinations of colors of Figure 4.6.*

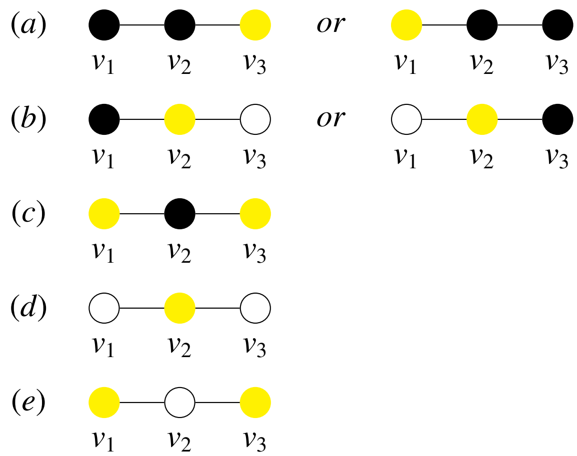

Figure 4.6: Possible valid colorings of dominating induced P_3

Proof. Clearly, by Corollaries 4.3 and 4.4 G has a dominating induced P_3. By Observations 4.1, 4.2 and 4.3, there exists neither two adjacent white vertices, nor yellow vertices with at least two non-white neighbors nor white vertices with black neighbors. As a consequence, the possible colors of the dominating induced P_3 must match to some combinations of Figure 4.6 or some of Figure 4.7. If the combination (f) of Figure 4.7 is matched then all vertices of the dominating induced P_3 are black vertices and by Observation 4.7, P is a trivial perfect edge dominating set leading to a contradiction. If the combination (g) of Figure 4.7 is matched, two adjacent vertices of the dominating induced P_3 are yellow vertices and the other one is a white vertex. None of these three vertices can have black neighbors which implies $B = \emptyset$ and P to be an efficient dominating set. Again, this is a contradiction. Therefore, the only valid options are those of Figure 4.6. □

4.3.3 Robust linear algorithm for P_5-free graphs

In this section, we describe a robust linear time algorithm for finding minimum weight perfect edge dominating set of a P_5-free graph. Given an arbitrary graph G, in linear time

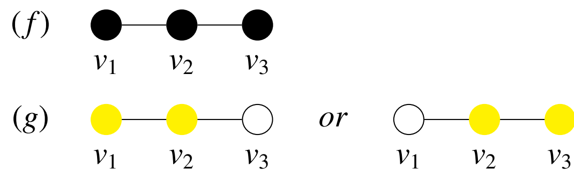

Figure 4.7: Invalid colorings of dominating induced P_3

the algorithm either finds such edge dominating set or exhibits an induced P_5 of the graph.

Let G, $|V(G)| > 1$, be an arbitrary connected graph. Along the process, the proposed algorithm constructs a set $\mathcal{E}$ containing a few candidates for the minimum weight perfect edge dominating set of G. At the end, it selects the least of them and returns the minimum edge dominating set of G,

Algorithm for minimum weighted perfect edge dominating set

1. Define $\mathcal{E} := \{E(G)\}$
2. Find a principal vertex of G. If such vertex does not exist, then return an induced P_5 and stop.
3. Using the principal vertex v, find (i) an induced P_5, or (ii) a dominating K_p, or (iii) a dominating P_3.
4. Case (i): An induced P_5 is found. Then return it and stop.
5. Case (ii): A dominating K_p is found. If $p \geq 4$ then return $E(G)$ and stop. Otherwise, $p \leq 3$ and using such dominating K_p, find a minimum weight efficient edge dominating set and, if it exists, include it in $\mathcal{E}$. If $K_p = \{v_1, v_2\}$ and $N(v_1) \cap N(v_2) = \emptyset$ then transform the dominating K_2 into a dominating P_3 by adding a third vertex to it.
6. Case (iii): A dominating P_3 is found. First, again using such a vertex dominating set, find a minimum weight efficient edge dominating set of G, and include it in $\mathcal{E}$, if it exists. Then look for a proper perfect edge dominating set of G, by considering

every possible coloring of the P_3, according to Figure 4.6. In each of the cases (a)-(b) ((c)-(e)), below, the algorithm determines at most two (one) proper perfect edge dominating sets (set), which are (is) then included in $\mathcal{E}$.

(a) Without loss of generality, v_3 is the yellow vertex of the dominating induced P_3. Clearly, $W = N(v_3) \setminus \{v_2\}$ and we can apply the linear-time algorithm of Theorem 4.7 to determine the proper perfect edge dominating set whose associated 3-coloring is (B, Y, W). Include it in $\mathcal{E}$, if it exists.

(b) Without loss of generality, v_1 is the black vertex of the dominating induced P_3. We can applied the same technique of (a) using $W = N(v_2) \setminus \{v_1\}$.

(c) Again, apply the same technique of (a) using $W = (N(v_1) \cup N(v_3)) \setminus \{v_2\}$.

(d) In this case, v_1 and v_3 have only yellow neighbors. As we are looking for proper perfect edge dominating sets, there is some black vertex somewhere. Clearly, v_2 must have exactly one black neighbor and the other neighbors of v_2 are white vertices. So, $Y = N(v_1) \cup N(v_3)$, $|B| = 1$, and we can apply the linear-time algorithm of Theorem 4.8 to determine the proper perfect edge dominating set whose associated 3-coloring is (B, Y, W) with least weight. Again, if successful include it in $\mathcal{E}$.

(e) In this case, v_1 (v_3) can have at most one black neighbor. As we are looking for proper perfect edge dominating sets, there is at least one black vertex. First, we assume that there are some triangles using edges of the dominating induced P_3 (the number of these triangles is exactly $|N(v_1) \cap N(v_2)| + |N(v_2) \cap N(v_3)|$ which can be computed in linear-time). If there are at least 2 triangles using the same edge, without loss of generality $\{v_1, v_2, x\}$ and $\{v_1, v_2, x'\}$, then the coloring is invalid. If $\{v_1, v_2, x\}$ and $\{v_2, v_3, x'\}$ are triangles then the coloring is invalid or it cannot admit black vertices, which means there are no proper perfect edge dominating sets. The only possibility for the graph to admit a proper perfect edge dominating set (which implies the existence of black vertices) is the existence of exactly one triangle using an edge of the dominating induced P_3 and $|B| = 1$. Clearly, the yellow vertices are $Y = N(v_2)$ and we then proceed as in Case (d). Next, suppose there is no triangle using an edge of the dominating induced P_3. Clearly, every vertex in $A_{1,3} = N(v_1) \setminus N(v_3)$ has to be adjacent to every vertex in $A_{3,1} = N(v_3) \setminus N(v_1)$ or there is some induced P_5, which the algorithm returns and then stops. It is not hard to see this can be accomplished in linear-time. Suppose that there is only a black vertex b. It has to be adjacent to v_1 and v_3, otherwise (without loss of generality, $bv_1 \notin E(G)$) there is an induced P_5 formed by y, v_1, v_2, v_3, b where y is a yellow neighbor of v_1. Then the yellow vertices are $Y = N(v_2)$ and we proceed as in (d). The last case is that there are two black vertices b_1, b_3 where $b_1 \in A_{1,3}$ and $b_3 \in A_{3,1}$, and the coloring is invalid if $|A_{1,3}| > 1$ or $|A_{3,1}| > 1$. The algorithm will check if $(B = \{b_1, b_3\}, Y = N(v_2), W = V \setminus (B \cup Y))$ is a valid coloring, and include in $\mathcal{E}$ the corresponding edge dominating set. All these computations can be done in linear-time.

7. Select the least weight perfect edge dominating set of $\mathcal{E}$, return it and stop.

The correctness and linear time complexity of the algorithm follows directly from the propositions formulated in the previous subsections.

To summarize we state the main result of this section.

Theorem 4.11. *The weighted perfect edge dominating set problem can be solved for P_5-free graphs in linear time in a robust way.*

Resumen del Capítulo 4

En este capítulo estudiamos la complejidad del problema del conjunto perfectamente dominante de aristas para algunas subclases de grafos. Si bien el enfoque de este capítulo es principalmente algorítmico, las observaciones y rutinas auxiliares tienen relevancia independiente y podrían ser usados para otras aplicaciones.

Asímismo queremos destacar que nuestras contribuciones forman parte de Lin et al. [56] y en este capítulo también mencionamos otros resultados de ese trabajo.

En la Sección 4.1 de este capítulo damos las definiciones necesarias e introducimos un 3-coloreo para manejar el problema.

Un subconjunto $E' \subset E(G)$ es un *conjunto perfectamente dominante de aristas* si toda arista de $E(G) \setminus E'$ es adyacente a exactamente una arista en E'. Por otro lado si toda arista de $E(G)$ es adyacente a exactamente una arista en E', entonces E' es un *conjunto eficientemente dominante de aristas*.

El *problema de cardinalidad* de la dominación perfecta de aristas consiste en determinar el conjunto perfectamente dominante de aristas de menor tamaño. El correspondiente *problema pesado* se define reemplazando la condición de menor tamaño por mínima suma de los pesos de las aristas dominantes.

En este capítulo distinguimos tres tipos de dominación perfecta de aristas:

- *eficiente*, cuando el conjunto dominante es también eficientemente dominante;
- *trivial*, cuando el conjunto dominante es $E(G)$;
- *propio*, cuando el conjunto dominante no cae en ninguna de las opciones anteriores.

En la Sección 4.2 estudiamos la complejidad del problema como parte del trabajo de Lin y otros [56]. En ese manuscrito se encuentra la prueba de la NP-completitud del problema cuando está restringido a los grafos sin garras de grado menor o igual que 3 (grafos subcúbicos). Y, en consecuencia, un interesante resultado sobre la dicotomía (polinomial o NP-difícil) de la complejidad del problema para grafos definidos con un subgrafo prohibido. En la Subsección 4.2.1 mostramos que la versión pesada del problema es lineal para la clase de grafos sin garras de grado exactamente 3 (grafos cúbicos), Teorema 4.3. Además, este resultado puede extenderse a la clase de grafos sin garras de grado a lo sumo 3 donde cada vértice de grado 2 tiene sus dos vecinos adyacentes. Esto muestra que cuando restringimos el problema del conjunto perfectamente dominante a la grase de grafos sin garras, la complejidad depende fuertemente del grado de los vértices.

En la Sección 4.3 estudiamos las clases de grafos sin P_5 motivados por el Corolario 4.1 de [56], que deja como problema abierto establecer la complejidad del problema para grafos definidos por un subgrafo prohibido H cuando H es un bosque lineal (unión de caminos). Notar que cualquier bosque lineal F contiene un P_k inducido y además F es un subgrafo inducifo de algún $P_{k'}$, $k' \geq k$. En este sentido, hicimos el trabajo para $k = 5$.

En la Subsección 4.3.1 presentamos observaciones y rutinas auxiliares para trabajar con los grapos P_5-free de manera robusta: Teoremas 4.5 y 4.6. Este último resultado constituye una prueba algorítmica del resultado de Bacsó y Tuza [2].

En la Subsección 4.3.2 enunciamos propiedades que relacionan los posibles 3-coloreos de subgrafos dominantes y la existencia de los distintos tipos de conjuntos perfectamente dominantes. El principal resultado en este sentido es el Teorema 4.10.

Finalmente, en la Subsección 4.3.3 describimos un algoritmo lineal y robusto para hallar un conjunto de mínimo peso perfectamente dominante de aristas para grafos que no contienen P_5.

Chapter 5

Vertex domination and well-covered graphs

A natural relation between vertex domination and edge domination, as we mentioned previously, is throughout line graphs. Recall that the line graph of a graph G, denoted by $L(G)$, is the graph with vertex set $E(G)$, where x and y are adjacent in the line graph if and only if edges x and y are adjacent in the original graph. Line graphs allows to make important connections between many important areas of graph theory. For example, determining a maximum matching in a graph is equivalent to finding a maximum independent set in the corresponding line graph. Similarly, edge coloring is equivalent to vertex coloring in the line graph. Much research has been done on the study and application of line graphs; a comprehensive survey is found in [75].

It is well known that the line graph can be obtain from the original graph in linear time. On the other direction, the *determination problem* consists in determining the original graph from its line graph. Whitney in [84] solved this problem by showing that with the only exception of $K_{1,3}$ and K_3, a graph is uniquely characterized by its line graph.

Our main objective in this thesis is to study edge domination. We can consider a very simple algorithm to find an edge dominating set which is just take any maximal matching. The greedy algorithm consists in adding edges to a matching until the matching becomes maximal and, clearly, ends in linear time. It is well known that a minimum maximal matching is a minimum edge dominating set. In the special case that all maximal matchings of the graph have the same size, the algorithm always determines a minimum edge dominating set. This is equivalent to say that, in the line graph, every maximal independent set has the same size. Graphs having all maximal independent sets of the same size are known as well-covered graphs. Therefore, there is a linear-time algorithm to solve vertex dominating set problem for well-covered line graphs.

In this chapter, Section 5.2 is dedicated to the complexity of vertex dominating set problem for well-covered graphs and its subclasses. We prove that the dominating set problem is a NP-complete for well-covered $K_{1,4}$-free graphs. Additionally, in Section 5.3 we address the recognition problem for certain subclasses of well-covered graphs related with comparability graphs. Since recognition of general well-covered graphs is coNP-complete, any polynomial time non-robust algorithm that solves vertex dominating set problem for some subclass of well-covered graphs is useful only if the recognition of such subclass can be done in polynomial time.

5.1 Preliminaries and previous results

Well-covered graphs were introduced by Plummer [72] in 1970. In the original definition, a graph is *well-covered* when every minimal vertex cover is also minimum. A vertex cover S satisfies that $I = V(G) \setminus S$ does not contain edges, then I is an independent set. Moreover, I is a maximal (maximum) independent set if and only S is a minimal (minimum) vertex cover. Therefore an equivalent definition for well-covered graphs is: a graph is *well-covered* if every maximal independent set is also maximum independent set. In other words, when all maximal independent sets are of the same size.

An *independent dominating set* is a dominating set of G that is also an independent set of vertices. Note that the independent dominating sets are exactly the maximal independent sets.

Recall that the *independent dominating number*, $i(G)$, is the minimum size of an independent dominating set; and the *independent number*, $\alpha(G)$, is the maximum size of an independent set of G. By definition, the inequality $i(G) \leq \alpha(G)$ holds. And we have $i(G) = \alpha(G)$ for well-covered graphs.

The problem of recognizing whether a graph is well-covered is coNP-complete [24, 79]. That means that deciding if a given graph is not well-covered is NP-complete; usually by showing two maximal independent set of different sizes. The recognition problem remains coNP-complete restricted to well-covered $K_{1,4}$-free graphs [20]. On the other hand, the recognition problem is polynomial for the following subclasses of well-covered graphs: claw-free graphs [81, 82], perfect graphs of bounded clique size [30], and graphs with bounded degree [21, 87], these last results also work for the weighted version of the problem. Many of the research in this area deals with characterization of some families of well-covered graphs. For instance, well-covered bipartite graphs [77], well-covered graphs with girth at least 5 [36], well-covered $\{C_4, C_5\}$-free graphs [37], well covered simplicial, chordal and circular arc graphs [76]. In some cases these characterizations leads to efficient algorithm for recognizing well-covered graphs.

The dominating set problem was proved to be NP-complete for well-covered graphs by Sankaranarayana and Stewart [79], and for line graph by Yannakakis and Gavril [40].

On the other hand, Allan and Laskar proved that every claw-free graphs have a minimum dominating set that is also independent [1]. Therefore in a well-covered claw-free graph any minimum independent dominating set, i.e. minimum maximal independent set, is a minimum dominating set. Using the very simple algorithm that adds vertices to an independent set until it becomes maximal, we can solve dominating set problem in polynomial time for well-covered claw-free graphs. This result implies that the dominating set problem is also polynomial for well-covered line graphs, since it is a subclass of well-covered claw-free graphs.

5.2 Vertex dominating set problem

In this section we prove that the dominating set problem is NP-complete for well-covered $K_{1,4}$-free graphs.

It is clear that the problem is in NP, because given a set of k vertices D we can verify in polynomial time whether D dominates the graph. In order to prove the NP-completeness, we use a version of the SAT problem called the 3-SAT$_{2+1}$ problem which is also NP-complete [71] and reduce our problem from it.

3-SAT$_{2+1}$ PROBLEM: Given variables $X_1, X_2, ..., X_n$ and a boolean formula written in a conjunctive normal form $C = C_1 \wedge C_2 \wedge ... \wedge C_m$, having 2 or 3 literals per clause, where each literal is an occurrence of some variable X_i or its negation $\neg X_i$; determine if it is possible to assign values *true* or *false* to the variables, in such a way that the formula is true. In this restricted version each variable occurs at most 3 times, twice positive and once negative.

5.2.1 Well covered $K_{1,4}$-free graphs

The following theorem also implies the NP-completeness of the dominating set problem restricted to well-covered $K_{1,4}$-free graphs with maximum degree $\Delta \leq 8$.

Theorem 5.1. *The dominating set problem is NP-complete for well-covered $K_{1,4}$-free graphs.*

Proof. We already mentioned that the problem is in NP. We shall prove that it is NP-complete by describing a reduction from 3-SAT$_{2+1}$.

Let be $C = C_1 \wedge C_2 \wedge ... \wedge C_k$ an instance of the 3-SAT$_{2+1}$ problem with variables $X_1, ..., X_n$. Define a graph G_C as follow:

- For each variable X_i define a triangle T with vertices $z_i, x_i, \neg x_i$.
- For each clause C_j define a graph L as in Figure 5.1 with distinct vertex c_j.
- For each clause $C_j = Y_{j1} \vee Y_{j2} \vee Y_{j3}$, where each literal can be X_i or $\neg X_i$, add edges between c_j and the vertices corresponding to Y_{j1}, Y_{j2}, and Y_{j3}, say y_{j1}, y_{j2}, and y_{j3}.
- Add edges between the vertices corresponding to y_{j1}, y_{j2}, and y_{j3} belonging to the same clause.

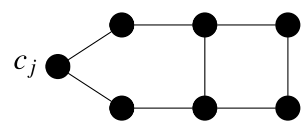

Figure 5.1: Graph L with distinct vertex c_j

Clearly, the graph G_C has $3n + 7k$ vertices, maximum degree 8, and can be computed in polynomial time. See an example in Figure 5.2.

Claim 1. *C is satisfiable if an only if G_C has a dominating set of size $n + 2k$.*

Proof: (⇒) Vertices corresponding to *true* literals in a satisfying truth assignment dominate all the copies of triangle T and vertices c_j, $j = 1, ..., k$. The remaining vertices can be dominated by two additional vertices for each copy of L. Therefore, a dominating set of size $n + 3k$ is obtained.

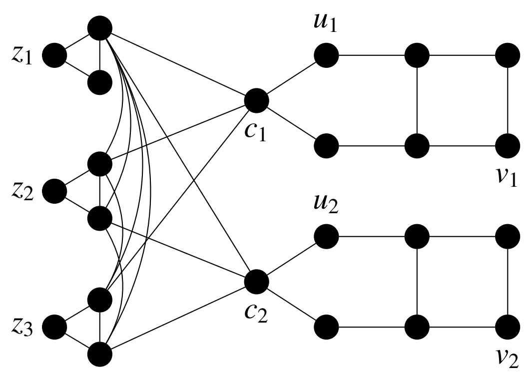

Figure 5.2: Example of the graph G_C defined by the imput $C = (X_1 \vee X_2 \vee X_3) \wedge (X_1 \vee \neg X_2 \vee \neg X_3)$

($\Leftarrow$) Any dominating set must contains at least one vertex from each copy of T since each z_i, $i = 1, ..., n$, is not adjacent to any vertex outside the triangle T. In addition, any dominating set must contain at least two vertices from each copy of L since vertices u_j, v_j, $j = 1, ..., k$, have disjoint neighborhoods. Therefore, a dominating set D of size $n + 2k$ contains exactly one vertex from each T and exactly two vertices from each L. Note that the distinct vertices c_j of the L subgraphs must all be dominated by vertices in the T triangles; otherwise, more than two dominating vertices would be required from some L. Thus the variable connectors which are in the dominating set correspond to the true literals of a satisfying truth assignment. This completes the proof of the claim. □

Claim 2. *G_C is a well-covered $K_{1,4}$-free graph.*

Proof: Clearly there is an induced $K_{1,4}$ if and only if there is a vertex v with $\alpha(G[N(v)]) \geq 4$. By construction the only vertices of degree ≥ 4 are c_j and possibly x_i and $\neg x_i$.

Vertices c_j, $j = 1, ..., k$, satisfies $\alpha(G[N[c_j]]) \leq 3$ since $N[c_j] \setminus \{u_j, v_j\}$ is complete. Each vertex $\neg x_i$, $i = 1, ..., n$, appears in at most one clause, say C_j, and then satisfies that $\alpha(G[N[\neg x_i]]) \leq 2$ since $N[\neg x_i] \cap N[c_j]$ is complete. Analogous, x_i $i = 1, ..., n$, appears in at most two clauses and then satisfies that $\alpha(G[N[x_i]]) \leq 3$.

We conclude that G_C is $K_{1,4}$-free.

To see that G is well-covered, let I be a maximal independent set. Clearly, $|I \cap N[z_i]| = 1$ for $1 \leq i \leq n$. In addition, I has exactly 3 vertices in each copy of L indistinctly if c_j belongs to the independent set or not. We have that $|I| = n + 3k$ for every maximal independent set and G_C is well-covered. □

This claim completes the proof of the theorem. □

5.3 Recognition problem

We have already discuss that the recognition of well-covered graphs is coNP-complete even under substantial restrictions. In this section we study the problem for well-covered comparability and well-covered co-comparability graphs. In both cases, we present polynomial time recognition algorithms.

Comparability graphs

A *comparability graph* is a graph that admits a transitive orientation of its edges. This is, each edge uv is oriented as $\overrightarrow{uv}$ or $\overleftarrow{uv}$, and whenever oriented edges $\overrightarrow{uv}$ and $\overrightarrow{vw}$ exists then edge uw also exists and has orientation $\overrightarrow{uw}$.

For the recognition problem of comparability graphs, McConnell and Spinrad developed a linear time algorithm to find a transitive orientation if it exists [65]. However deciding if a given orientation is transitive requires $O(MM(n))$ (matrix multiplication). Morvan and Viennot [68] presented parallel algorithms for the recognition problem and the computation of a transitive orientation of comparability graphs; their algorithms run in $O(\log n)$ time and require $O(\Delta m)$ processors on the CRCW PRAM model, where Δ is the maximum degree of a vertex in the graph.

A transitive orientation of the edges defines a partial ordering R of the vertices V by defining $u < v$ if and only if the edge uv is oriented $\vec{uv}$. We also say that v is a predecessor of u and that u is a successor of v. Note that G is the undirected graph of the directed (acyclic) graph $G(V, R)$.

Following observations arise straightforward from the definition of a transitive orientation.

Observation 5.1. *Vertices in a directed path defines a complete set in G.*

Observation 5.2. *Every edge of G either (i) belongs to a maximal directed path or (ii) can be deduced by transitivity from a maximal directed path.*

By Observation 5.1, every clique in a comparability graph corresponds with a maximal directed path in the directed graph $G(V, R)$.

A vertex is called a *maximum* when it has not predecessors, and a *minimum* when it has not successors. In polynomial time, we can find a reduced graph G' with the same vertex set than G and edges consisting in the essential directed edges of G. If necessary, transitivity can be used to transform G' back to the original graph.

5.3.1 Well-covered comparability graphs

We use a equivalent definition of well-covered graphs described by Dean and Zito in [30]. We present the case of their theorem that we need for our objectives, the complete version appears in the cited work.

Theorem 5.2. *[30] The following statements are equivalent:*

(a) G is well-covered.

(b) For every clique C there is no nonmaximum independent set S of such that (i) $S \cap C = \emptyset$, (ii) $|C| \geq |S|$, and (iii) $C \subseteq N(S)$.

Lemma 5.1. *If a vertex set S with $S \cap C = \emptyset$ dominates a clique in a comparability graph C, then there are two vertices in S that also dominates C.*

Proof. Let $p_1, p_2, ..., p_k$ be the maximal directed path defined by C and oriented by $\overrightarrow{p_j p_{j+1}}$, $j = 1, ..., k-1$. Consider $a \in S$ such that exists an oriented edge $\overrightarrow{p_i a}$ with maximum i, and $b \in S$ such that exists an oriented edge $\overrightarrow{b p_j}$ with j minimum. By transitivity, a dominates vertices $p_1, p_2, ..., p_i$, and b dominates $p_j, p_{j+1}, ..., p_k$. By definition of i and j, vertices p_h with $i < h < j$, are not adjacent to any vertex in S. Since S dominates C, we obtain $j \leq i + 1$ and $\{a, b\}$ dominates C. Note that a and b can not be the same vertex, otherwise $a \in C$, and may be not unique. □

Theorem 5.3. *There is a polynomial algorithm to determine whether a given graph is a well-covered comparability graph.*

Proof. We describe the algorithm for a connected graph, if the input is not connected apply the algorithm in each connected component. Given a graph G, find a transitive orientation and determine if G is a comparability graph. Next, determine if G satisfies statement (b) of Theorem 5.2 in the following way. By Lemma 5.1 it is sufficient to verify (b) for independent sets of size 2. Note that if G is connected then $|C| \geq 2$ and condition (ii) is always satisfied. Since G is well-covered, S is a nonmaximum independent set if and only if is nonmaximal independent, and this is equivalent to $N[S] \neq V(G)$.

Statement (b) is equivalent to say for each pair of non adjacent vertices a, b such that $N(a) \cup N(b) \neq V(G)$, there does not exist maximal paths of G contained in $N(a) \cup N(b)$. These verifications can be done in polynomial time. □

5.3.2 Well-covered co-comparability graphs

The co-comparability graphs are the complement of comparability graphs. The recognition problem of co-comparability graphs is polynomial time solvable since so is deciding if $\bar{G}$ is a comparability graph.

We will see in this section a simple proof that the problem of recognize well-covered co-comparability graph is polynomial time solvable. Note that the complexity of dominating set problem is also polynomial for these graphs [54].

Theorem 5.4. *There is a polynomial time algorithm to decide if a given graph is well-covered co-comparability graph.*

Proof. As we have seen in the previous section, a transitive orientation of edges in $\bar{G}$ can be obtained and verified in $O(MM(n))$. We can also check in linear time if all maximal cliques in $\bar{G}$ have same cardinality as follows.

By Observation 5.1 a clique in $\bar{G}$ is a maximal directed path. After finding all sources S and terminals T in the directed acyclic graph $\bar{G}$. In linear time we can determine a shortest and a longest path between S and T. If these paths have the same length all cliques in $\bar{G}$ have the same size.

In this situation, all maximal independent sets of G are of the same size and G is well-covered. □

Resumen del Capítulo 5

En este capítulo estudiamos la relación entre la dominación de vértices y de aristas que surge naturalmente a través de los grafos de líneas. Recordemos que el grafo de línea de un grafo G, notado $L(G)$, tiene como vértices a las aristas de G, donde x e y son adyacentes en $L(G)$ si y sólo si las aristas x e y son adyacentes en G.

Podemos considerar un algoritmo simple para encontrar un conjunto dominante de aristas, que es seleccionar cualquier matching maximal. El algoritmo goloso consiste en ir agregando aristas a un matching hasta que el matching sea maximal; claramente este algoritmo termina en tiempo lineal. Se sabe que un matching maximal mínimo es un conjunto dominante de aristas de tamaño mínimo. Si además estamos en el caso que todos los matching maximales tienen el mismo tamaño, el algoritmo goloso siempre determina un conjunto dominante de aristas mínimo. Esta condición es equivalente a decir que, en el grafo de línea, todos los conjuntos independientes maximales tienen el mismo tamaño. Los grafos donde todos los conjuntos independientes maximales tienen el mismo tamaño se llaman grafos *bien cubiertos*. Por lo tanto hay un algoritmo lineal para resolver el problema de conjunto dominante de vértices para grafos bien cubiertos.

En la Sección 5.1 damos algunos resultados relacionados. Entre ellos mencionamos la prueba de la NP-completitud del problema de conjunto dominante para grafos bien cubiertos [79] y para grafos de línea [40]. Por otro lado, en [1] se prueba que todo grafo sin garras contiene un conjunto dominate mínimo que además es independiente. Entonces, el algoritmo goloso resuelve el problema del conjunto dominante mínimo para grafos bien cubiertos sin garras ($K_{1,3}$).

En la Sección 5.2 probamos que el problema del conjunto dominante es NP-difćil para grafos bien cubiertos sin $K_{1,4}$, (Teorema 5.1).

En la Sección 5.3 discutimos el problema de reconocimiento de grafos bien cubiertos para subclases relacionadas con grafos de comparabilidad. Notar que el reconocimiento de los grafos bien cubiertos es un problema coNP-completo [24, 79]. En consecuencia, cualquier algoritmo polinomial no robusto que resuelva el problema de dominación de vértices para alguna subclase de grafos bien cubiertos es útil sólo si el reconocimiento de dicha subclase se puede hacer en tiempo polinomial.

Un grafo es de *comparabilidad* si es posible orientar sus aristas de manera transitiva. Esto significa que cada arista uv se orienta como $\overrightarrow{uv}$ o $\overleftarrow{uv}$, y si existen las aristas orientadas $\overrightarrow{uv}$ y $\overrightarrow{vw}$ entonces también existe la arista uw y tiene orientación $\overrightarrow{uw}$.

Un grafo es de *co-comparabilidad* si su grafo complemento, $\bar{G}$, es de comparabilidad.

Los resultados son dos algoritmos polinomiales, uno para el reconocimiento de grafos bien cubiertos de comparabilidad (Teorema 5.3) y otro para los grafos bien cubiertos de co-comparabilidad (Teorema 5.4).

Chapter 6

Conclusions

In this thesis, we have made contributions on several dominating-set related problems. The analysis of subgraphs and graph parameters related to vertex domination and edge domination gives a better insight in the relation between different versions of the problems. In addition, we analyzed the complexity and provided new efficient algorithms for these problems.

In Chapter 2, we studied edge dominating set problem and give an algorithm that solves the problem for proper interval graphs. (Theorem 2.2)

The efficient edge dominating set (DIM) problem is addressed in Chapter 3. The coloring of the vertices used in this chapter allows us to manage the problem in a simple way, and to identify the structures related to the existence of many DIMs. This approach leads to tight bounds for the maximum number of DIMs in the following graph classes: general (Theorem 3.2), triangle-free (Theorem 3.3) and connected (Theorem 3.4). Also, the characterization of the extremal graphs naturally arise. These results were published in the Journal of Graph Theory (2015) [61], and have been presented in a communication of the Annuals of the Unión Matemática Argentina (2015).

The perfect edge domination is more general than efficient edge domination. In Chapter 4 we use a 3-coloring of the vertices to represent the solutions of the problem. Then, different types of perfect edge dominating sets (trivial, efficient, and proper) can be easily identified, as well as their relation with some induced subgraphs and dominating subgraphs. Exploring these structural properties we provided efficient algorithms for some graph classes. Firstly, we prove that the weighted perfect edge dominating set problem is linear-time solvable for cubic claw-free graph (Theorem 4.3) and it can be extended easily for a more general class. Secondly, we proved in Theorem 4.11 that the weighted perfect edge dominating set problem is linear-time solvable for P_5-free graphs in a robust way. These results are part of the manuscript [56] which is submitted to a journal.

We leave as an open problem the question whether or not there exists some graph class for which the efficient edge dominating set problem is NP-complete and the perfect edge dominating set problem can be solved in polynomial time.

Motivated by the relation between vertex and edge domination several graph classes were been studied in Chapter 5. Some properties of well-covered graphs and comparability graphs are useful to identify certain subclasses of graphs where the vertex dominating set problem and recognition problem are polynomial-time solvable. We stated the NP-completeness of the dominating set problem for well-covered $K_{1,4}$-free graphs (Theorem 5.1). This result determines the threshold for the complexity in well-covered graphs, since dominating set problem is polynomial when restricting to well-covered $K_{1,3}$-free graphs. The analysis of the properties also leads to linear-time algorithms for the recognition problem of well-covered comparability graphs (Theorem 5.3) and well-covered co-comparability graphs (Theorem 5.4).

We leave as an open question if there exists a subclass of well-covered graphs such that:

(i) recognition problem is polynomial and dominating set problem is NP-complete, or

(ii) recognition problem is coNP-complete and dominating set problem is polynomial?

Maybe well-covered comparability graphs are good candidates for (i).

To summarize, we present a list of our contribution.

- A linear time algorithm that solves the edge dominating set problem for proper interval graphs;
- new tight bounds for the number of dominating induced matchings for the class of general graphs, triangle-free graphs and connected graphs;
- the characterization of the extremal graphs for the studied classes;
- a linear time algorithm for the perfect edge dominating set problem for cubic claw-free graphs, and an extension of this algorithm for subcubic graphs where every degree-two vertex has two adjacent neighbors;
- a robust linear time algorithm that solves weighted perfect edge dominating set problem for P_5-free graphs;
- a NP-Completeness proof for vertex dominating set problem restricted to well-covered $K_{1,4}$-free graphs;
- a polynomial time algorithm that solves the recognition problem for well-covered comparability graphs;
- a polynomial time algorithm that solves the recognition problem for well-covered co-comparability graphs.

Conclusiones

En esta tesis hemos hecho aportes en varios problemas relacionados al conjunto dominante. El análisis de subgrafos y parámetros relacionados a la dominación de vértices y aristas permiten una mejor comprensión de la relación entre las diferentes versiones de los problemas. Adicionalmente, analizamos la complejidad y damos nuevos algorithmos eficientes para resolver estos problemas.

En el Capítulo 2, estudiamos el problema del conjunto dominante de aristas y damos un algoritmo que resuelve el problema para los grafos de intervalos propios, Teorema 2.2.

El problema del conjunto eficientemente dominante de aristas (problema DIM) es abordado en el Capítulo 3. El coloreo de los vértices usado en este capítulo nos permite manejar el problema de manera simple e identificar las estructuras relacionadas con la existencia de muchos DIMs. Esta estrategia nos conduce a cotas ajustadas para el máximo número de DIMs de un grafo en las siguientes clases de grafos: clase general (Teorema 3.2), grafos sin triángulos (Teorema 3.3) y grafos conexos (Teorema 3.4). Adicionalmente, las caracterizaciones de los grafos extremales surgen naturalmente. Estos resultados fueron publicados en el Journal of Graph Theory (2015) [61], y han sido presentados en una comunicación en la Reunión Anual de la Unión Matemática Argentina (2015).

La dominación perfecta de aristas es más general que la dominación eficiente. En el Capítulo 4 usamos un 3-coloreo de los vértices del grafo para representar las soluciones del problema. Luego, los diferentes tipos de conjuntos perfectamente dominantes (trivial, eficientes y propios) pueden ser fácilmente identificados, así como su relación con algunos subgrafos inducidos y subgrafos dominantes. Explorando estas propiedades estructurales, damos algoritmos eficientes para algunas clases de grafos. En primer lugar, probamos que la versión pesada del problema del conjunto perfectamente dominante se puede resolver en tiempo linear para los grafos cúbicos sin garras (Teorema 4.3) y además se puede extender fácilmente a una clase más general. En segundo lugar, probamos que la versión pesada del problema del conjunto perfectamente dominante se puede resolver en tiempo lineal para los grafos sin P_5 de manera robusta (Teorema 4.11). Estos resultados forman parte del manuscrito [56] que fue enviado a una revista para su publicación.

Dejamos como una pregunta abierta si existe o no alguna clase de grafos para la cual el problema de la dominación eficiente sea NP-completo mientras que el problema de la dominación perfecta se pueda resolver en tiempo polinomial.

Motivados por la relación entre la dominación de vértices y de aristas, distintas clases de grafos son estudiadas en el Capítulo 5. Algunas propiedades de los grafos bien cubiertos y grafos de comparabilidad son útiles para identificar subclases de grafos donde el problema del conjunto dominante de vértices y el problema de reconomiento se resuelven en tiempo polinomial. En el Teorema 5.1 establecemos la NP-completitud del problema del conjunto dominante para los grafos bien cubiertos sin $K_{1,4}$. Este resultado determina un umbral para la complejidad dentro de los grafos bien cubiertos, ya que el problema del conjunto dominante es polinomial cuando está restringido a los grafos bien cubiertos sin $K_{1,3}$. El análisis de las propiedades también conduce a algoritmos de tiempo lineal para el problema de reconocimiento de grafos bien cubiertos de comparabilidad (Teorema 5.3) y grafos bien cubiertos de co-comparabilidad (Teorema 5.4).

Dejamos como pregunta abierta la existencia de una subclase de grafos bien cubiertos para la cual:

(i) el problema de reconocimiento sea polinomial y el problema de conjunto dominante sea NP-completo, o

(ii) el problema de reconocimiento sea coNP-completo y el problema de conjunto dominante sea polinomial?

Quizás los grafos bien cubiertos de comparabilidad sean buenos candidatos para (i).

A modo de síntesis, presentamos una lista de nuestros aportes.

- Un algoritmo lineal que resuelve el problema de dominación de aristas para grafos de intervalos propios;
- nuevas cotas ajustadas para el número de matching inducidos dominantes para la clase general de grafos, grafos sin triángulos y grafos conexos;
- las caracterizaciones de los grafos extremales para las clases estudiadas;
- un algortirmo de tiempo lineal para el problema de la dominación perfecta de aristas para grafos cúbicos sin garras, y una extensión de este algoritmo para los grafos subcúbicos donde todo vértice de grado 2 tiene dos vecinos adyacentes;
- un algoritmo robusto de tiempo lineal que resuelve el problema pesado de la dominación perfecta de aristas para grafos sin P_5;
- una prueba de la NP-completitud del problema del conjunto dominante de vértices restringido a los grafos bien cubiertos sin $K_{1,4}$;
- un algoritmo de tiempo polinomial que resuelve el problema de reconocimiento de grafos bien cubiertos de comparabilidad;
- un algoritmo de tiempo polinomial que resuelve el problema de reconocimiento de grafos bien cubiertos de co-comparabilidad.

Bibliography

[1] R.B. Allan, and R. Laskar, On domination and independent domination numbers of a graph, Discrete Mathematics. 23 (1978) 73-76. 90, 97

[2] G. Bacsó, Zs. Tuza, Dominating cliques in P_5-free graphs, Periodica Mathemayica Hungarica, 21 (1990) 303-308. 74, 78, 86

[3] C. Berge, Les problèmes de colorations en théorie des graphes. Publ. Inst. Stat. Univ. Paris, 9 (1960) 123-160. 12, 22

[4] C. Berge, Färbung von Graphen deren sämtliche bzw. deren ungerade Kreise starr sind (Zusammenfassung), Wissenschaftliche Zeitschrift, Martin Luther Universität Halle-Wittenberg, Mathematisch-Naturwissenschaftliche Reihe (1961) 114-115. 12, 22

[5] C. Berge, Theory of Graphs and its Applications. Methuen, London, (1962). 14, 24

[6] N. Biggs, E. Lloyd and R. Wilson, Graph Theory 1736-1936, Clarendon Press, (1976). 13, 23

[7] H. L. Bodlaender and M. A. Langston, editors, Proceedings 2nd International Workshop on Parameterized and Exact Computation, IWPEC 2006, Springer Verlag, Lecture Notes in Computer Science, vol. 4169 (2006). 111

[8] B. Bollobfis and E.J. Cockayne, Graph-theoretic parameters concerning domination, independence and irredundance, Journal of Graph Theory, 3 (1979) 241-249.

[9] K.S. Booth, G.S. Lueker, Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. Journal of Computer and System Sciences. 13 (3) (1976) 335-379. 33, 41

[10] A. Brandstädt, C. Hundt, and R. Nevries, Efficient edge domination on hole-free graphs in polynomial time, Lecture Notes in Computer Science, 6034 (2010) 650-661. 46

[11] A. Brandstädt, V.B. Le, and J.P. Spinrad, Graph Classes: A Survey. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (1999). 13, 14, 23, 24

[12] A. Brandstädt, A. Leitert, and D. Rautenbach, Efficient dominating and edge dominating sets for graphs and hypergraphs, Lecture Notes in Comput. Sci. 7676 (2012) 267-277. 46

[13] A. Brandstädt and R. Mosca, Dominating induced matching for P_7-free graphs in linear time, Lecture Notes in Comput. Sci. 7074 (2011) 100-109.

[14] A. Brandstädt and R. Mosca, Finding dominating induced matchings in P_8-free graphs in polynomial time, R. Algorithmica. (2016) doi:10.1007/s00453-016-0150-y. 46

[15] B. Brešar, P. Dorbec, W. Goddard, B. Hartnell, M. Henning, S. Klavžar, D. Rall, Vizing's Conjecture: A survey of Recent Results, Journal of Graph Theory, 69 (1) (2012) 46-76. 12, 22

[16] E. Camby, O. Schaudt, A new characterization of P_k-free graphs, *Graph-Theoretic Concepts in Computer Science - 40th International Workshop (WG 2014)*, France, *Revised Selected Papers*, (2014) 129-138. 76

[17] D.M. Cardoso, J.O. Cerdeira, C. Delorme, and P.C. Silva, Efficient edge domination in regular graphs, Discrete Applied Mathematics. 156 (2008) 3060-3065. 45, 46, 65

[18] D.M. Cardoso, N. Korpelainen, and V.V. Lozin, On the complexity of the dominating induced matching problem in hereditary classes of graphs, Discrete Applied Mathematics. 159 (2011) 521-531. 46

[19] D.M. Cardoso and V.V. Lozin, Dominating induced matchings, Lecture Notes in Comput. Sci. 5420 (2009) 77-86. 45

[20] Y. Caro, A. Sebö, and M. Tarsi, Recognizing greedy structures, J. Algorithms 20 (1996), 137-156. 89

[21] Y. Caro, M. N. Ellingham, and J.E. Ramey, Local structure when all maximal independent sets have equal weight, SIAM Journal of Discrete Mathematics, 11 (1998) 644-654. 89

[22] R. Carr, T. Fujito, G. Konjevod, and O. Parekh, A 21/10 approximation algorithm for a generalization of the weighted edge-dominating set problem, Journal of Combinatorial Optimization, 5 (2001) 317-326. 31

[23] G. Chartrand, M. Jacobson, editors. Surveys in Graph Theory. Congressus Numerantium 116 (1996). 115

[24] V. Chvátal, and P.J. Slater, A note on well-covered graphs, in 'Quo Vadis, Graph Theory?', Annals of Discrete Mathematics. 55 (1993) 179-182. 17, 27, 89, 98

[25] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas, The strong perfect graph theorem. Annals of Mathematics 164 (1) (2006) 51-229. 12, 22

[26] E.J. Cockayne and S.T. Hedetniemi, Towards a theory of domination in graphs. Networks 7 (1977) 211-219. 12, 14, 22, 24

[27] S.A. Cook, The complexity of theorem-proving procedures. In [45], (1971) 151-158. 12, 22

[28] V. Dahllödtf, and P. Jonsson, An algorithm for counting maximum weighted independent sets and its applications. In [32], (2002) 292-298. 46

[29] C. F. de Jaenisch, Applications de l'Analyse mathematique an Jen des Echecs, (1862). 14, 24

[30] N. Dean and J. Zito, Well-covered graphs and extendability, Discrete Mathematics. 126 (1994) 67-80. 89, 94

[31] R. G. Downey and M. R. Fellows, Fixed-parameter tractability and completeness, Congressus Numerantium, 87 (1992) 161-178. 31

[32] D. Eppstein, editor. Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms, January 6-8, 2002, San Francisco, CA, USA. ACM/SIAM. (2002). 110

[33] P. Erdös, A.W. Goodman, and L. Pósa, The representation of a graph by set intersections, Canadian Journal of Mathematics. 18 (1966) 106-112. 13, 23

[34] U. Feige, A threshold of ln n for approximating set cover, Journal ACM, 45 (1998) 634-652. 31

[35] H. Fernau, Edge dominating set: Efficient enumeration-based exact algorithms. In [7] (2006) 140-151. 31

[36] A. Finbow, B. Hartnell, and R.J. Nowakowski, A characterization of well-covered graphs of girth 5 or greater, Journal of Combinatorial Theory, Ser B 57 (1993) 44-68. 89

[37] A. Finbow, B. Hartnell, and R.J. Nowakowski, A characterization of well-covered graphs that contain neither 4- nor 5-cycles, Journal of Graph Theory, 18 (1994) 713-721. 89

[38] M.R. Garey, and D.S. Johnson. Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA. (1990). 14, 17, 18, 24, 27, 28

[39] F. Gavril, Algorithms on circular-arc graphs, Networks. 4, 4 (1974) 357-369. 13, 23

[40] F. Gavril, and M. Yannakakis, Edge dominating sets in graphs. SIAM Journal of Discrete Mathematics. 38, 3, (1980) 364-372. 18, 28, 31, 33, 41, 89, 97

[41] J. P. Georges, M. D. Halsey, A. M. Sanaulla, M. A. Whittlesey, Edge domination and graph structure, *Congressus Numerantium* 76, (1990) 127-144. 67

[42] W. Goddard, and M. Henning, Independent domination in graphs: A survey and recent results, Discrete Mathematics. 313 (2013) 839-854. 14, 24

[43] D.L. Grinstead, P.J. Slater, N.A. Sherwani, and N.D. Holmes, Efficient edge domination problems in graphs, Inform. Process. Lett. 48 (1993) 221-228. 43, 45, 46, 64

[44] F. Harary, Graph theory, Addison-Wesley, Reading, MA, 1969. 34

[45] M.A. Harrison, R.B. Banerji, and J.D. Ullman, editors. Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, May 3-5, 1971, Shaker Heights, Ohio, USA. ACM. (1971). 110

[46] T.W. Haynes, S.T. Hedetniemi, and P.J. Slater, editors. Domination in Graphs: Advanced Topics, Marcel Dekker, New York, (1998). 14, 24

[47] M. A. Henning, A survey of selected recent results on total domination in graphs, Discrete Mathematics. 309 (2009) 32-63. 14, 24

[48] M. A. Henning, and A. Yeo, Total domination in graphs, Springer Monographs in Mathematics (2013). 14, 24

[49] A. Hertz, V. Lozin, B. Ries, V. Zamaraev, D. de Werra, Dominating induced matchings in graphs containing no long claw, arXiv:1505.02558. 46

[50] S.F. Hwang, and G.J. Chang, The edge domination problem, Manuscript. 33

[51] J.D. Horton, and K. Kilakos, Minimum edge dominating sets, SIAM Journal of Discrete Mathematics. 6 (1993) 375-387. 33

[52] J. G. Kalbfleisch, R. G. Stanton and J. D. Horton, On covering sets and error-correcting codes. J. Comb. Theory 11A (1971) 233-250. 12, 22

[53] N. Korpelainen, A polynomial-time algorithm for the dominating induced matching problem in the class of convex graphs, Electron. Notes Discrete Mathematics. 32 (2009) 133-140. 46

[54] D. Kratsch, and L. Stewart, Domination on cocomparability graphs SIAM Journal of Discrete Mathematics. 6(3) (1993) 400-417. 96

[55] C. Lekkeikerker, and J. Boland, Representation of a finite graph by a set of intervals on the real line. Fundamenta Mathematicae, 51(1) (1962) 45-64. 13, 23

[56] M.C. Lin, V.V. Lozin, V. Moyano, and J.L. Szwarcfiter, Perfect Edge Domination: Hard and Solvable Cases. Manuscript. 20, 29, 68, 70, 71, 73, 85, 86, 100, 104

[57] M.C. Lin, M.J. Mizrahi, and J.L. Szwarcfiter, An $O^*(1.1939^n)$ time algorithm for minimum weighted dominating induced matching, Lecture Notes in Comput. Sci. 8283 (2013) 558-567. 46

[58] M. C. Lin, M. Mizrahi, J. L. Szwarcfiter, Exact algorithms for dominating induced matching. *Corr*, abs/1301.7602, (2013). 79

[59] M. C. Lin, M. Mizrahi, J. L. Szwarcfiter, Fast algorithms for some dominating induced matching problems, *Information Processing Letters* 114, (2014) 524-528. 71, 72, 73

[60] M. C. Lin, M. Mizrahi, J. L. Szwarcfiter, Efficient and perfect domination on circular-arc graphs, *Proceedings of the VIII Latin-American Graphs, Algorithms and Optimization Symposium (LAGOS' 2015)*, Beberibe, Brazil, *Electronic Notes in Discrete Mathematics*, to appear (2015). 68

[61] M. C. Lin, V. Moyano, D. Rautenbach, J. L. Szwarcfiter, The Maximum Number of Dominating Induced Matchings, *Journal of Graph Theory* 78, (2015) 258-268. 99, 103

[62] M. Livingston and Q.F. Stout, Distributing resources in hypercube computers, In Proc. 3rd Conf. on Hypercube Concurrent Computers and Applications (1988), ACM, 222-231. 43

[63] C.L. Lu, M.-T. Ko, and C.Y. Tang, Perfect edge domination and efficient edge domination in graphs, Discrete Applied Mathematics. 119 (2002) 227-250. 46, 67, 68

[64] C.L. Lu and C.Y. Tang, Solving the weighted efficient edge domination problem on bipartite permutation graphs, Discrete Applied Mathematics. 87 (1998) 203-211. 46

[65] R.M. McConnell, J. Spinrad, Linear-time transitive orientation, 8th ACM-SIAM Symposium on Discrete Algorithms (1997) 19-25. 93

[66] T.A. McKee, Topics in intersection graph theory. (1999). 16, 26

[67] S. Mitchell, and S. Hedetniemi, Edge Domination in Trees, Congr. Numer. 19 (1977) 489-509. 33

[68] M. Morvan, L. Viennot, Parallel comparability graph recognition and modular decomposition. In [74], (1996) 169-180. 93

[69] P. Nagavamsi, Domination and related problems on some classes of perfect graphs, Tech. project report, Dept. of Comp. Sci. and Engr. IIT Madras (1993). 33

[70] O. Ore, Theory of graphs, American Math. Society Colloquium Publications, 38 (1962). 14, 24

[71] C.H. Papadimitriou, Computational Complexity, Addison Wesley, (1994). 90

[72] M.D. Plummer, Some covering concepts in graphs, Journal of Combinatorial Theory, 8 (1970) 91-98. 88

[73] A. Poghosyan, and U. of the West of England, The Probabilistic Method for Upper Bounds in Domination Theory. PhD thesis. University of the West of England, (2010). 14, 24

[74] C. Puech, and R. Reischuk, STACS 96: 13th Annual Symposium on Theoretical Aspects of Computer Science Grenoble, France, February 22-24, 1996 Proceedings. Springer Berlin Heidelberg (1996). 114

[75] E. Prisner, Line Graphs and Generalizations. In [23], (1996) 193-229. 87

[76] E. Prisner, J.Topp, and P. D. Vestergaard, Well-covered simplicial, chordal, and circular arc graphs, Journal of Graph Theory, 21 (1996) 113-119. 89

[77] G. Ravindra, Well-covered graphs, Journal of Combinatorics, Information and System Sciences. 2 (1977) 20-21. 89

[78] W. W. Rouse Ball, Mathematical Recreation and Problems of Past and Present Times, (1892). 14, 24

[79] R. S. Sankaranarayana and L. K. Stewart, Complexity results for well covered graphs, Networks 22 (1992), 247-262. 17, 27, 89, 97, 98

[80] A. Srinivasan, K. Madhukar, P. Nagavamsi, C. Pandu Rangan, and M.S. Chang, Edge domination on bipartite permutation graphs and cotriangulated graphs, Inform. Process. Lett., 56 (1995), pp. 165-171. 33

[81] D. Tankus, and M. Tarsi, Well covered claw-free graphs, Journal of Combinatorial Theory, Ser B 66 (1996) 293-302. 89

[82] D. Tankus, and M. Tarsi, The structure of well-covered graphs and the complexity of their recognition problems, Journal of Combinatorial Theory, Ser B 69 (1997) 230-233. 89

[83] V.G. Vizing, The Cartesian product of graphs, Vyčisl. Sistemy, 9 (1963) 30-43. 12, 22

[84] H. Whitney, Congruent Graphs and the Connectivity of Graphs, American Journal of Mathematics, 54 (1932) 150-168. 87

[85] M. Xiao, H. Nagamochi, Exact algorithms for dominating induced matching based on graph partition, Discrete Applied Mathematics, 190-191, (2015) 147-162. 46

[86] A. M. Yaglom and I. M. Yaglom. Challenging mathematical problems with elementary solutions. Volume 1: Combinatorial Analysis and Probability Theory, (1964). 14, 24

[87] I.E. Zverovich, Weighted well-covered graphs and complexity questions, Moscow Mathematical Journal, 4 (2004) 523-528. 89