



HAL
open science

Extension of NP Optimization Problems

Mehdi Khosravian Ghadikolaei

► **To cite this version:**

Mehdi Khosravian Ghadikolaei. Extension of NP Optimization Problems. Computational Geometry [cs.CG]. Université Paris sciences et lettres, 2019. English. NNT : 2019PSLED064 . tel-03220653

HAL Id: tel-03220653

<https://theses.hal.science/tel-03220653>

Submitted on 7 May 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à Université Paris Dauphine

Extension of NP Optimization Problems

Soutenue par

Mehdi KHOSRAVIAN
GHADIKOLAEI

Le 19/07/2019

École doctorale n°543

ED de Dauphine

Spécialité

Informatique

Composition du jury :

| | |
|---|---------------------------|
| Mamadou Moustapha KANTÉ Maître de Conférences, Université Clermont Auvergne | <i>Rapporteur</i> |
| David MANLOVE Professeur, University of Glasgow | <i>Rapporteur</i> |
| Marthe BONAMY Maître de Conférences, Université de Bordeaux | <i>Examineur</i> |
| Bruno ESCOFFIER Professeur, Sorbonne Université | <i>Président de jury</i> |
| Michel HABIB Professeur, Université Paris Diderot | <i>Examineur</i> |
| Martin MILANIČ Maître de Conférences, University of Primorska | <i>Examineur</i> |
| Jérôme MONNOT Directeur de recherche, Université Paris Dauphine | <i>Directeur de thèse</i> |
| Florian SIKORA Maître de Conférences, Université Paris Dauphine | <i>Examineur</i> |

Acknowledgement

First of all, I would like to express my gratitude to my thesis advisor Professor Jérôme Monnot for allowing me to carry out this thesis. Thank you for everything you have contributed, scientifically throughout this work. For always letting your door wide open to answer my questions and help me overcome whatever hardships came to my way. For your friendship and your undeniable support. For all the things that you have taught me during these years in your domain of interest, in computational complexity, approximability, etc... I would also like to thank my second advisor Dr. Florian Sikora for all of his contributions throughout this thesis. For setting me on the right path when I lost sight of what laid ahead, for his friendship, patience motivation, and guidance both on a professional and personal level. For all of our joint meetings and varied discussions.

Besides, I would like to thank Dr. Mamadou Moustapha Kanté from Université Clermont Auvergne, for accepting to review my thesis and act as a reporter. It is my honor to have you as part of the scientific committee members. Thank you for attending my pre-defense session and proposing guidance lines so as to improve my work. I was very honored that Professor David Manlove from the University of Glasgow has agreed to report my thesis. I started my Ph.D. program by studying his excellent Ph.D. thesis, that inspired me and gave me a plethora of ideas for conducting my research.

It is also my pleasure to have Dr. Marthe Bonamy from Université de Bordeaux, Professor Bruno Escoffier from Sorbonne University, Professor Michel Habib from Université Paris Diderot and Dr. Martin Milanič from the University of Primorska in the scientific committee members of my thesis. My sincere thanks go to you for accepting to thoroughly review my thesis, for your insightful comments and suggestions.

I would also like to thank Mr. Henning Fernau, Professor at Trier University in Germany, Dr. Katrin Casel from Hasso Plattner Institute and Dr. Kaveh Khoshkakh from Tartu University in Estonia for all of your contributions to this thesis. It was my pleasure to have exchanged with you, thank you for our scientific discussions and intelligent remarks.

My humble gratitude goes to the faculty members of the LAMSADE lab for all of their scientific support, friendship and encouragement. Particularly, I would like to thank Professor Vangelis Paschos, Dr. Michael Lampis, Dr. Eun Jung Kim, Dr. Ararat Harutyunyan, and Dr. Denis Cornaz for their inspiring courses and research that helped me along the way.

I would also like to thank all my colleagues and friends from the LAMSADE with whom I have shared unforgettable moments in recent years. Thank you for the lively and heated discussions, for your friendship, for all the work carried out together and all the time we got to spend in laughter and joy.

It is hard to find the words to express my gratitude to all my outside-the-university friends for their friendship, their support and the important role they play in my life. Finally, my acknowledgement would not be complete if I do not thank my family, who have supported me throughout my life and who have always been an inspiration and an infinite source of motivation. I would like to especially thank my darling wife Yalda, without who, this work would not have seen the light and would not have been possible to accomplish, thank you dear for your support, constant presence, for believing in me and for the courage you always brought me.

Résumé

Le problème de la détermination de la qualité d'une solution partielle se pose dans presque toutes les approches algorithmiques qui calculent progressivement une solution globale. L'élagage des arbres de recherche, la preuve des garanties d'approximation et l'efficacité des stratégies d'énumération sont des approches algorithmiques qui exigent souvent un moyen approprié de décider si une solution partielle donnée est un candidat raisonnable à l'extension à une solution globale, de qualité assurée. Dans cette thèse, nous considérons cette exigence d'"extensibilité" dans le contexte des problèmes d'optimisation de graphes.

Nous étudions un type particulier de problèmes d'optimisation, appelés *problèmes d'extension* pour la classe des problèmes de graphes *héréditaires* et *anti-héréditaires*. Dans la version d'extension d'un problème Π de graphe anti-héréditaire ou héréditaire, nous considérons une instance $G = (V, E)$ de Π avec un ensemble supplémentaire spécifié de pré-solution U (qui est un sous-ensemble de E ou V , selon le problème) et le but est de faire de la pré-solution une solution *minimal* (ou *maximal*) de valeur optimum.

Nous considérons un grand nombre de problèmes d'optimisation sur les graphes dans le cadre des problèmes d'extension. Contredisant peut-être l'intuition, ces problèmes ont tendance à être NP-difficile, même quand le problème d'optimisation sous-jacent peut être résolu en temps polynomial. Nous présentons de nombreux résultats positifs et négatifs de NP-difficulté et d'approximation pour différents scénarios d'entrée. De plus, nous étudions la complexité paramétrée des problèmes d'extension par rapport à la taille des pré-solutions, ainsi que l'optimalité de certains algorithmes exacts sous l'hypothèse de temps exponentielle.

Parfois, il n'y a pas de solution réalisable à un problème d'extension, ce qui rend la question de l'existence de telles extensions intéressante. Ainsi, *la version de décision d'un problème d'extension* pour un problème de graphe Π anti-héréditaire (ou héréditaire), peut être formulé comme suit: Étant donné un graphe G et une pré-solution U , le problème consiste à trouver une solution minimale (resp. maximale) S tel que $S \supseteq U$ (resp. $S \subseteq U$). À cet égard, nous étudions quelques problèmes classiques de graphes tels que le problème de *couverture par sommets*, *de stable*, *d'ensemble dominant*, *de couplages* et de *couverture par arêtes*. Nous présentons aussi quelques résultats positifs et négatifs de NP-difficulté et de complexité paramétrée. Nous considérons des scénarios très restreints, tels que des restrictions de degré ou de topologie, ainsi que la complexité paramétrée par rapport à la taille de la pré-solution et à la largeur d'arborescence du graphe. Considérant la possibilité que certains ensembles U ne soient pas extensibles à une solution minimale, on pourrait se demander si U est un mauvais choix pour un problème d'extension. Pour cela, nous avons défini le concept du *prix de l'extension* (PoE), et proposé quelques résultats d'inapproximation.

Nous étudions également quelques problèmes de graphes dans le cadre de *minimum-maximal* et *maximum-minimal*, qui est un cas particulier des problèmes d'optimisation naturels liés au PoE, lorsque la pré-solution U est vide ou lorsque $U = V$ (resp. $U = E$ selon le problème). Enfin, nous proposons un cadre général basé sur les ordres partiels.

Abstract

The problem of determining the quality of a partial solution occurs in almost every algorithmic approach that gradually computes a global solution. Pruning search trees, proving approximation guarantees, or the efficiency of enumeration strategies usually require a suitable way to decide if a given partial solution is a reasonable candidate to pursue for extension to a global one, of assured quality. In this thesis, we consider this “extendibility” requirement for efficient computation, especially in the context of graph optimization problems.

We study a special type of optimization problems, called *extension problems* for the class of *hereditary* and *anti-hereditary* graph problems. Informally, in the extension version of an anti-hereditary (resp. a hereditary) graph problem Π , we consider an instance $G = (V, E)$ of Π with an additionally specified set of pre-solutions U (which is a subset of E or V , depending on the problem), while the goal is to *extend* the pre-solution to an (inclusion-wise) *minimal* (resp. *maximal*) one of optimum value.

We consider a large number of optimization problems on graphs within the framework of extension problems. Possibly contradicting intuition, these problems tend to be NP-hard, even for problems where the underlying optimization problem can be solved in polynomial time. We present many positive/negative hardness and approximation results for different input scenarios. Moreover, the parameterized complexity of extension problems with respect to the size of the pre-solutions, as well as the optimality of some exact algorithms under the Exponential-Time Hypothesis (ETH) are studied.

Sometimes there is no feasible solution for an instance of an extension problem which makes the question of the existence of such extensions interesting. Hence, the *decision extension problem* for an anti-hereditary (resp. a hereditary) graph problem Π , is given in advance a graph G together with a pre-solution U , and it is asked whether a minimal (resp. maximal) solution S exists such that $S \supseteq U$ (resp. $S \subseteq U$). In this regard, we investigate some classical graph problems such as *vertex cover*, *independent set*, *dominating set*, *edge matching* and *edge cover*, etc, and present some positive and negative hardness and parameterized results. All complexity considerations took place in very restricted scenarios, such as degree or topological restrictions, as well as in a parameterized setting with respect to the size of pre-solution along with the treewidth of the given graph. Considering the possibility that some set U might not be extendable to any minimal solution, one might ask how wrong U is as a choice for an extension problem. We defined the concept *price of extension* (PoE), and find some (in-)approximability results in this regard.

We also study some graph problems in the framework of *minimum-maximal* and *maximum-minimal*, which is a particular case of the natural optimization problems related to PoE in our framework, when the pre-solution is *empty-set* or *whole-set*. Finally, we propose a general framework based on partial-ordering.

Résumé étendu

Ce manuscrit retrace une étude générale du comportement des problèmes d'optimisation dans les graphes avec une solution partielle donnée, plus généralement connus sous le nom de *problèmes d'extension*. De nombreux problèmes d'extension sont étudiés dans la littérature – de tels problèmes impliquent généralement l'optimisation d'une fonction objectif dans l'espace des solutions réalisables contenant une solution partielle ou excluant d'autres éléments. Pour les problèmes d'extension contenant une solution partielle (également appelés problèmes de complétion), il existe de nombreux résultats concernant les problèmes de coloration [23, 109], de planification [84], de routage [7, 8, 113] ainsi que d'autres problèmes combinatoires examinés dans [64, 161]. Une recherche approfondie a été menée sur le fameux problème d'EXTENSION DE PRÉ-COLORATION (EN ANGLAIS PRECOLORING EXTENSION) dans différentes classes de graphes. Dans ce problème, on considère un graphe et une coloration propre partielle donnés en entrée et on cherche à l'étendre à l'ensemble du graphe. [139, 109, 146, 14, 159, 138, 23]. À titre d'exemple, dans les articles de [139, 109], il a été démontré que ce problème est NP-difficile dans les graphes d'intervalles propres lorsque le nombre de couleurs de la solution partielle n'est pas borné. Toujours à titre d'exemple, le problème PRECOLORING EXTENSION est polynomial dans graphes d'intervalles lorsque chaque couleur n'apparaît qu'une seule fois dans la solution partielle mais il devient NP-difficile dès lors qu'une couleur de l'ensemble pré-coloré est utilisée deux fois [23]. Pour les graphes bipartis, le problème est NP-difficile si l'ensemble pré-coloré comprend 3 sommets de couleurs distinctes.

Une variante d'extension du problème du PLUS COURT CYCLE (SHORTEST CYCLE) consiste en un graphe pondéré $G = (V, E, w)$ et un chemin $U \subseteq (E \cup V)$, le but étant d'étendre U à un cycle simple ayant un poids minimum. Bien que ce problème soit NP-difficile, il s'avère FPT pour le paramètre naturel qui est le nombre de sommets ou arêtes dans le cycle le plus court [24]. Dans le cadre des problèmes d'extension, une version généralisée du PROBLÈME DU VOYAGEUR DE COMMERCE (TRAVELLING SALESMAN PROBLEM (TSP)) connu sous le nom du PROBLÈME GÉNÉRAL DE ROUTAGE (GENERAL ROUTING PROBLEM (GRP)) a été considéré du point de vue de l'approximation. Étant donné un graphe $G = (V, E)$ et deux sous-ensembles $V' \subseteq V$, $E' \subseteq E$, l'objectif du GRP est de trouver un cycle de coût minimum qui passe par tous les sommets de V' et qui contient chaque arête de E' . Dans [113], à l'aide de l'algorithme de Christofides pour le TSP avec inégalité triangulaire, les auteurs présentent une $\frac{3}{2}$ -approximation pour le GRP. En outre, le PROBLÈME DU CLUSTER ORDONNÉ DU VOYAGEUR DE COMMERCE (ORDERED CLUSTER TRAVELING SALESMAN (OCTSP)) a également été étudié dans la littérature [7, 8, 91, 99]. Nous considérons comme entrée dans ce problème un graphe pondéré $G = (V, E, w)$ et une partition ordonnée des sommets de G en k groupes C_1, \dots, C_k . Le but de l'OCTSP est de trouver un tour simple passant par tous les sommets et ayant un poids minimum de telle sorte que l'ordre des sommets soit respecté, en d'autres termes, le tour visite d'abord tous les sommets en C_1 , puis tous les sommets en C_2, \dots , et enfin tous les sommets du cluster C_k . D'autres types d'extension s'intéressent aux problèmes d'optimisation dans des graphes de forçage ou de conflit [62], dans lesquels, pour une instance donnée du problème du chemin le plus court, une contrainte pour une paire de sommets distincts $\{u, v\}$ est appelée une contrainte de conflit (resp. forçage) si au maximum (resp. Au moins) un

sommet parmi u ou v est traversé par ce chemin.

Dans cette thèse, de nombreux problèmes d'optimisation héréditaires et anti-héréditaires sont étudiés dans le cadre des problèmes d'extension. Formellement, un problème d'optimisation $\Pi = \langle I, F, opt, m \rangle$ est **héréditaire** (resp. **anti-héréditaire**) si pour toute instance $x \in I$, l'ensemble des solutions réalisables $F(x)$ est fermé pour l'inclusion d'ensemble (resp. l'exclusion), c'est-à-dire pour toute instance $x \in I$ et toute solution réalisable $y \in F(x)$ $y' \in F(x)$ pour chaque $y' \subseteq y$ (resp. $y' \supseteq y$). Dans notre cas, l'instance x représente un graphe $G = (V, E)$ et $F(G) \subseteq 2^V$ ou $F(G) \subseteq 2^E$ selon le contexte. Par exemple, MAXIMUM INDEPENDANT SET est un problème héréditaire de graphe, car dans un graphe $G = (V, E)$, l'ensemble $S' \subseteq S$ est indépendant si $S \subseteq V$ est un ensemble indépendant, tandis que MIN VC est un problème anti-héréditaire car pour un graphe donné $G = (V, E)$, $S' \supseteq S$ est une couverture par sommets de G si $S \subseteq V$ l'est aussi.

Dans le cadre de notre étude, nous nous concentrons sur des solutions minimales et maximales en termes d'inclusion. Étant donné un problème d'optimisation de graphes $\Pi = \langle I, F, opt, m \rangle$, pour une instance $G \in I$ et un ensemble de solutions réalisables $F(G)$, l'ensemble des solutions réalisables minimales (resp. Maximales) en termes d'inclusion est désigné par $\mu(G)$ (resp. $\psi(G)$) et défini comme suit :

- $\mu(G) = \{y \in F(G) : ((y' \subseteq y) \wedge (y' \in F(G))) \rightarrow y' = y\}$
- $\psi(G) = \{y \in F(G) : ((y' \supseteq y) \wedge (y' \in F(G))) \rightarrow y' = y\}$

Soit Π un problème d'optimisation anti-héréditaire (resp. héréditaire). Pour une instance $G \in I$ de Π , une instance du problème d'extension Π' inclut G ainsi qu'une pré-solution U . Cette dernière représente un sous-ensemble d'un élément de l'entrée correctement choisi en fonction du problème considéré. Par exemple, pour le problème MINIMUM VERTEX COVER, chaque instance contient un graphe $G = (V, E)$ et un sous-ensemble $U \subseteq V$, alors que pour le problème MAX EM, l'instance inclut un graphe $G = (V, E)$ et une solution partielle $U \subseteq E$. Ainsi, lorsqu'il s'agit d'un problème anti-héréditaire Π (resp. un problème héréditaire), la pré-solution U , est appelée un ensemble forcé. (resp. un ensemble co-interdit).

Dans ce chapitre, nous présentons une définition formelle de tout type de problème d'optimisation ou de décision dans le cadre de l'extension. Nous essayons également de passer en revue toutes les études connexes dans chaque section. Nous clôturons un chapitre par un bref aperçu des résultats et des problèmes qui seront présentés dans les chapitres suivants.

Problèmes d'optimisation d'extension

Dans cette partie, nous définissons les problèmes d'optimisation de graphes liés aux problèmes d'extension. Soit Π un problème d'optimisation de graphe anti-héréditaire (resp. héréditaire). Le problème d'extension Π' noté par MIN EXT Π (resp. MAX EXT Π) se compose d'une instance G de Π alliant un **ensemble forcé (forced set)** (resp. un **ensemble co-interdit (a co-forbidden set)**¹ $U \subseteq V$ (ou E selon le contexte) comme entrée, et le but de Π' est de trouver une solution

¹ \bar{U} est appelé ensemble co-interdit(co forbidden set).

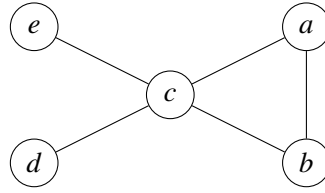


Figure 1: Le graphe $G = (V, E)$ comme une instance de MAX EXT IS et MIN EXT VC.

minimale (resp. maximale) S de sorte que $U \subseteq S$ (resp. $U \supseteq S$). Par exemple, pour les deux problèmes MIN VC et MAX IS, les variantes d'extension sont définies comme suit :

MIN EXT VC

Entrée: Un graphe $G = (V, E)$, un forced set $U \subseteq V$.

Solution: Une couverture par sommets minimale $S \subseteq V$ telle que $U \subseteq S$.

Sortie: $|S|$ minimisé.

MAX EXT IS

Entrée : Un graphe $G = (V, E)$, un co-forbidden set $U \subseteq V$.

Solution : un stable maximal $S \subseteq V$ telle que $S \subseteq U$.

Sortie : $|S|$ maximisé.

Il est important de noter que la minimalité et la maximalité sont considérées par rapport à l'ensemble des solutions réalisables. Ainsi, pour une instance (G, U) de MIN EXT VC, un sous-ensemble $S \supseteq U$ est une couverture par sommet minimale si $S - v$ n'est pas une couverture par sommet pour tout $v \in S$. Un autre problème d'extension associé à la couverture par sommet est défini par $S \supseteq U$, où S est une couverture par sommet et $S - v$ n'est pas une couverture par sommet pour tout $v \in S \setminus U$. Dans ce cas, nous limitons toutes les solutions réalisables pour contenir U , avec le but est d'optimiser la partie solution de U ; cette variante a été étudiée dans [161, 64]. Weller et al. ont défini une variante des problèmes d'extension, où, pour une solution réalisable S du problème, le but est d'optimiser la partie résiduelle $(S \setminus U)$ [161]. Ces auteurs ont étudié quelques problèmes classiques d'optimisation ainsi que la description de l'approximation résiduelle dans le cadre des problèmes d'extension. Dans [64], les auteurs ont ajouté une autre contrainte au problème, dans laquelle, pour une instance (G, U, F) d'un problème Π , toute solution possible S contient tous les membres de U tandis que $S \cap F = \emptyset$. De nombreux problèmes d'optimisation dans les graphes ont été étudiés dans ce cadre du point de vue de complexité exacte et de l'approximation. Concernant un autre type de problème d'extension considéré dans [57], pour un problème de graphe Π . Dans ce contexte, une instance consiste en un graphe $G = (V, E)$ et une partition Π_V (resp. Π_E) de ses sommets (resp. de ses arêtes), et toute solution S contenant un élément (sommets ou arête) de toute partition doit également contenir toutes les autres. Donnons maintenant un exemple illustrant le contexte de notre étude.

Exemple 0.1. Considérons le graphe présenté dans la Figure 1 comme exemple pour MIN EXT VC et MAX EXT IS. Supposons d'abord que G ainsi que l'ensemble forcé $U = \{a, b\}$ est une

instance de MIN EXT VC. La seule couverture par sommet minimale qui puisse contenir $\{a, b\}$, est $\{a, b, e, d\}$, par conséquent la taille de la solution optimale de MIN EXT VC pour l'instance (G, U) est 4. Cependant, dans le cadre de l'extension proposée par Weller et al. [161]. (en se concentrant uniquement sur la partie résiduelle), la couverture par sommets optimale contenant U est $\{a, b, c\}$. Il y a deux couvertures par sommets minimale de G contenant U , $\{a, b, d\}$ et $\{a, b, c, e, f\}$. Comme notre but est de trouver une solution réalisable de taille minimum, $\{a, b, d\}$ est la solution optimale. Supposons maintenant que le graphe G avec l'ensemble co-interdit $U = \{c, d, e, a\}$ est une instance de MAX EXT IS. Dans ce cas, il y a deux ensembles indépendants maximaux qui incluent U à savoir, $\{a, e, d\}$, et $\{c\}$. Ainsi, la taille de la solution optimale de MAX EXT IS pour l'instance (G, U) est de 3.

Dans ce qui suit, nous définissons formellement $\text{ext}_\Pi(G, U)$ pour les problèmes héréditaires et anti-héréditaires. Soit Π un problème de graphes anti-héréditaire (resp. héréditaire), $G = (V, E)$ est un graphe simple et $U \subseteq V$ ou E (selon le contexte), et $\text{ext}_\Pi(G, U)$ est un sous-ensemble $S \subseteq V$ or E (selon le contexte) de solutions minimales (resp. maximales) de Π telles que $S \supseteq U$ (resp. $S \subseteq U$). Soit Π un problème de graphe anti-héréditaire, soit $G = (V, E)$ un graphe simple et soit $U \subseteq V$ ou E (selon le contexte). Alors, $\text{ext}_\Pi(G, U)$ est un sous-ensemble $S \subseteq V$ ou E (selon le contexte) de solutions minimales de Π telles que $S \supseteq U$. Maintenant que Π est défini comme un problème de graphe héréditaire, où $G = (V, E)$ est un graphe simple et $U \subseteq V$ ou E (selon le contexte). Dès lors, $\text{ext}_\Pi(G, U)$ est un sous-ensemble $S \subseteq V$ or E (selon le contexte) de solutions maximales de Π telles que $S \subseteq U$.

Parfois, pour une instance (G, U) d'un problème d'extension Π , $\text{ext}_\Pi(G, U) = \emptyset$, en d'autres termes il n'existe pas de solution possible pour le problème, ceci suscite un intérêt particulier pour la question d'existence de ces extensions. Cependant, si l'on commence par une solution partielle triviale, il est généralement facile de trouver la réponse à cette question. Dans la section suivante, nous définissons les problèmes de décision associés aux problèmes abordés dans le cadre de cette thèse.

Problèmes de décision associés aux problèmes d'extension

La question de savoir si une solution partielle donnée à un problème peut être raisonnablement étendue a été développée dans de nombreuses approches algorithmiques destinées aux problèmes d'optimisation. Par exemple, en énumérant les ensembles dominants minimaux d'un graphe $G = (V, E)$, on arrive généralement au problème de décision dans lequel on cherche à savoir étant donné pour ensemble de sommets $U \subseteq V$, s'il existe un ensemble dominant *minimal* S avec $U \subseteq S$. Pour un problème héréditaire (resp. anti-héréditaire) donné Π , le *problème de décision d'extension* pour Π est dénoté $\text{EXT } \Pi$. De manière informelle, une entrée de $\text{EXT } \Pi$ consiste en une instance $G = (V, E)$ de Π , avec une pré-solution $U \subseteq V$ (ou E selon le contexte) et le problème de décision associé s'intéresse à l'existence d'une solution minimale (resp. maximale) S de sorte que $U \subseteq S$ (resp. $U \supseteq S$). Par exemple, pour les problèmes de couverture par sommets et de stable, les versions d'extension qui leur sont associées EXT VC et EXT IS sont formellement définies comme suit :

EXT VC

Entrée: Un graphe $G = (V, E)$ et un ensemble forcé $U \subseteq V$.

Question: Existe-t-il une couverture par sommet minimale S de telle sorte que $U \subseteq S$?

EXT IS

Entrée: Un graphe $G = (V, E)$ et un ensemble co-interdit $U \subseteq V$.

Question: Existe-t-il un stable maximal S telle que $U \supseteq S$?

Dans la Remark 4.1, nous montrerons que (G, U) est une instance positive d'EXT VC si et seulement si $(G, V \setminus U)$ est une instance positive du problème EXT IS. Pour clarifier, nous présentons un exemple ci-après.

Exemple 0.2. Considérons le graphe G illustré dans la Figure 2, et supposons que $(G, \{g, h, f\})$ et $(G, \{g, a, e\})$ sont deux instances de EXT VC. Il est clair que $\{g, h, f\}$ peut être étendu à une couverture par sommet minimale pour G , car $\{g, h, h, f, d, b\}$ est une couverture par sommet minimale contenant U . Par contre, $(G, \{g, a, e\})$ est une instance négative de EXT VC, c'est à dire qu'il n'existe aucune couverture par sommet minimale dans G qui peut contenir $\{g, a, e, e\}$. Supposons que $(G, \{a, h, c, f, e\})$ et $(G, \{b, h, c, f, d\})$ sont deux instances de EXT IS. Il est facile de montrer que $(G, \{a, h, c, f, e\})$ est une instance positive de EXT IS, car $\{a, c, e\}$ est un stable maximal. Toutefois, il n'existe pas de stable maximal inclus dans $\{b, h, c, f, d\}$, car au moins un des sommets $\{a, e, g\}$ doit être ajouté au stable de $\{b, h, c, f, d\}$. Celui-là appartient forcément à un ensemble indépendant plus grand, donc $(G, \{b, h, c, f, d\})$ est une non-instance de EXT IS.

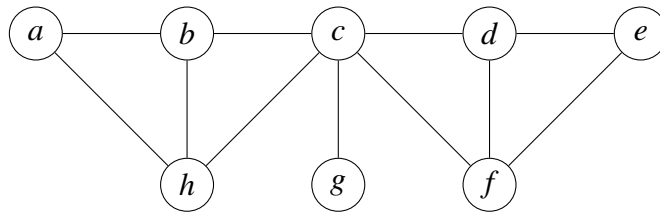


Figure 2: Le graphe $G = (V, E)$ comme une instance de EXT IS et EXT VC.

Ce type de problème de décision lié à un problème d'optimisation d'extension est rencontré dans de nombreux algorithmes d'énumération efficaces : lors de l'énumération de toutes les solutions minimales pour une instance d'un problème, il est généralement supposé qu'on dispose d'une pré-solution U , s'il existe une solution *minimale* S avec $U \subseteq S$. Ce problème de décision a été examiné dans de nombreuses études dans le cadre du dénombrement, par exemple, dans les travaux suivants: [31, 32, 33, 59, 80, 92, 93, 95, 120, 119, 142, 152]. Le problème d'extension du problème de couverture par sommet appelé MINIMAL VERTEX COVER EXTENSION (MVCEXT), consiste en un graphe $G = (V, E)$ et un ensemble $Y \subseteq V$ en entrée avec l'objectif de trouver tous les ensembles X jouissant de ces deux propriétés : (1) $X \supseteq Y$, et (2) X est une couverture par sommet minimale de G . Soit κ -MVCEXT le même problème avec une contrainte supplémentaire (3) $|X| \leq k$. Ces deux problèmes ont été étudiés dans [59]. Plus généralement, la question

de l'extension des solutions minimales s'est posée dans le contexte des preuves de complexité pour les algorithmes de dénombrement (efficace) des formules booléennes, des matroïdes ainsi d'autres cas similaires; voir [32, 120].

D'ailleurs, dans [32], il est démontré que l'extension des solutions partielles est NP-difficile pour calculer les principaux impliquants du dual d'une fonction booléenne; un problème qui peut aussi être vu comme une tentative pour trouver un ensemble intersectant minimal (hitting set) pour les principaux impliquants de la fonction d'entrée. Interprétée de cette façon, la preuve de [32] résulte en la preuve de la NP-difficulté pour le problème d'extension minimale pour 3-HITTING SET (bien que ce problème peut être résolu en temps polynomial quand $|U|$ est constant). Ce résultat a été étendu dans [17] pour prouver la NP-difficulté de calculer l'extension de sous-ensembles de sommets à des ensembles dominants minimaux (EXT DS), même limitées aux graphes planaire cubiques. De même, il a été montré dans [16] que les extensions des couvertures de sommets minimales limitées aux graphes cubiques planaires sont NP-difficile. La première étude *systématique* de ce type de problèmes a été présentée dans [41] fournissant un certain nombre d'exemples de ce type de problèmes.

Un *système indépendant* est un système défini comme suit: (V, \mathcal{E}) , $\mathcal{E} \subseteq 2^V$, qui est héréditaire pour l'inclusion. Le problème d'extension EXT IND SYS (aussi appelé FLASHLIGHT) du système indépendant a été proposé dans [131]. Dans ce problème, étant donné en entrée deux sous-ensembles $X, Y \subseteq V$, on cherche l'existence d'un stable maximal incluant X qui n'intersecte pas Y . Lawler et al. ont prouvé que EXT IND SYS est NP-complet, même si $X = \emptyset$ [131]. Afin d'énumérer tous les ensembles dominants minimaux (en termes d'inclusion) pour un graphe donné, Kanté et al. ont étudié une restriction de EXT IND SYS qui consiste à trouver un ensemble dominant minimal contenant X . Les auteurs ont prouvé que EXT DS est NP-complet, et ce même pour les classes de graphes spéciales comme les split graphs, les graphes cordaux et les line graphs [117, 116]. De plus, ils ont proposé un algorithme linéaire pour les split graphs lorsque X, Y est une partition de la partie clique du graphe [115].

Un *système indépendant* est un système défini comme suit: (V, \mathcal{E}) , $\mathcal{E} \subseteq 2^V$, qui est héréditaire pour l'inclusion. Le problème d'extension EXT IND SYS (aussi appelé FLASHLIGHT) du système indépendant a été proposé dans [131]. Dans ce problème, étant donné en entrée deux sous-ensembles $X, Y \subseteq V$, on cherche l'existence d'un stable maximal incluant X qui n'intersecte pas Y . Lawler et al. ont prouvé que EXT IND SYS est NP-complet, même si $X = \emptyset$ [131]. Afin d'énumérer tous les ensembles dominants minimaux (en termes d'inclusion) pour un graphe donné, Kanté et al. ont étudié une restriction de EXT IND SYS qui consiste à trouver un ensemble dominant minimal contenant X . Les auteurs ont prouvé que EXT DS est NP-complet, et ce même pour les classes de graphes spéciales comme les split graphs, les graphes cordaux et les line graphs [117, 116]. De plus, ils ont proposé un algorithme linéaire pour les split graphs lorsque X, Y est une partition de la clique. Le cas où l'ensemble forcé U satisfait certaines propriétés comme le fait d'être une clique ou un stable a déjà été étudié sous un certains aspects :

- **Clique Forte (Strong Clique)** Une clique forte (strong clique) dans un graphe est une clique qui intersecte tout stable maximal. Dans [107], Hujdurovic et al. ont étudié six problèmes de décision liés aux cliques fortes. Ils ont prouvé qu'une clique C dans un graphe G n'est

pas forte si et seulement si elle est dominée par un stable $I \subseteq V(G) \setminus C$. En d'autres termes, ils ont montré qu'une clique C est forte si et seulement s'il y a une couverture par sommet minimale de G s'étendant à C . De plus, en introduisant la variante d'extension du problème de la clique forte, les auteurs ont décrit une version généralisée de leurs problèmes. EXT CLIQUE prend en entrée un graphe G et une clique C de G , et cherche si G contient une forte clique $C' \supseteq C$. Un aperçu de la clique forte se trouve dans [M. Milanič, Strong cliques and strong stable sets in graphs, chapitre à paraître en: Topics in Algorithmic Graph Theory, Cambridge University Press, edited by L. Beineke, M. Golumbic, R. Wilson].

- **Stable disjoint (Disjoint Independent set)** Le problème de décision qui cherche si un graphe contient une paire de stables maximum et disjoints est un vieux problème dans la théorie des graphes qui a été introduit par Berge au début des années 70. Récemment, de nombreuses études ont été réalisées afin de trouver une paire de sous-ensembles disjoints maximum/minimum de graphes satisfaisant la même propriété [53, 65, 150, 108]. Par exemple, dans [105], les auteurs prouvent que le problème de décision qui cherche si un graphe a deux stables maximums disjoints est un problème NP-complet. En supposant que l'un des sous-ensembles disjoints est donné à l'avance, le problème se transforme en problème de décision d'extension. De cette version, on suppose qu'on a un graphe $G = (V, E)$ ayant un stable maximal $x \subseteq V$ connu a priori, et on demande si G a un stable maximal inclus dans $V \setminus X$.
- **Extension de représentation partielle (Extension of partial representation).** Dans le contexte de d'étendre une représentation partielle donnée en une représentation complète pour une sous-classe de graphes d'intersection, plusieurs résultats sont proposés dans [6, 124, 124, 124, 121, 123, 48]. Par exemple, dans [124, 122], le problème s'intéresse à la question de si la représentation partielle peut être étendue à une représentation du graphe entier et ce dans un graphe d'intervalle donné avec quelques intervalles pré-dessinés en entrée. Dans [122, 124], les auteurs proposent un algorithme linéaire pour étendre respectivement les représentations d'intervalles propre et des graphes d'intervalles en utilisant des arbres PQ. Dans [122], les auteurs ont également introduit un problème plus général de représentations bornées des graphes d'intervalles unitaires qui s'avère NP-complet, où l'entrée limite les positions de certains intervalles par des bornes inférieure et supérieure.

Prix de l'extension

Il est possible que certains ensembles U ne soient pas extensibles à une solution minimale ou maximale, une question triviale qui se pose tente de savoir à quelle point l'ensemble U se rapproche d'un ensemble extensible. Afin d'évaluer la distance séparant U d'un ensemble extensible, on peut estimer le nombre d'éléments de U qui doivent être modifiés pour avoir une solution minimale. Décrit de manière différente pour les problèmes d'extension que nous abordons, nous voulons discuter les méthodes de quantification qui permettent d'évaluer le

nombre d'éléments de U qui nécessitent d'être supprimés (resp. le nombre d'éléments de $X \setminus U$ qui doivent être ajoutés, où X est l'univers) pour un problème anti-héréditaire (resp. héréditaire), afin d'obtenir une instance positive en termes d'inclusion pour le problème d'extension. A cet égard, nous définissons une nouvelle notion de prix d'extension (Price of extension PoE), afin d'expliquer l'effet de la contrainte supplémentaire (sur la solution partielle) sur la possibilité de trouver des solutions minimales (resp. maximales). Une approche similaire a déjà été utilisée dans le passé sous le nom du *prix de connectivité* (*the Price of Connectivity*) dans [38] pour le contexte de la connectivité. Ceci est dû à son importance majeure dans les applications de routage; cette notion a été introduite dans [38] pour MIN VC et est définie comme le ratio maximum exprimé par le rapport entre le nombre de sommets constituant une couverture par sommets connectés et le nombre de sommets d'une couverture par sommets simple. Dans notre contexte, le but de la PoE est d'évaluer le rapprochement entre les sous-ensembles extensibles, calculables à partir de la solution partielle connue, sont proches de U ou des sous-ensembles extensibles les plus larges possibles obtenus à partir de U . Pour aborder formellement ce concept, nous présentons les problèmes d'optimisation suivants. À partir d'une instance (G, U) d'un problème anti-héréditaire (resp. héréditaire) Π , deux nouveaux problèmes d'optimisation dénotés par $\text{EXT}_{\max}\Pi$ (resp. $\text{EXT}_{\min}\Pi$) et définis comme suit :

$\text{EXT}_{\max}\Pi$

Entrée: Une instance x ainsi qu'un ensemble forcé U .

Solutions: Une solution minimale S de $G = (V, E)$.

Sortie: Une solution S qui maximise $|S \cap U|$.

$\text{EXT}_{\min}\Pi$

Entrée: une instance x avec un ensemble co-interdit U .

Solutions: Une solution Maximale S de x .

Sortie: Une solution S qui minimise $|S \cup U|$.

Pour éclaircir les définitions développées ci-avant, nous introduisant l'exemple suivant.

Example 0.3. Supposons que (G, U) est une instance de EXT_{\max} VC où $G = (V, E)$ est le graphe représenté dans la Figure 3 et $U \subseteq V = \{c, d, e\}$. On peut voir que (G, U) est une instance négative de EXT VC alors que les ensembles de sommets $\{c, d\}, \{e\}$ sont extensibles à une couverture par sommet minimale. Comme notre objectif est de trouver une couverture par sommet minimale contenant un nombre maximal de sommets de U , la solution optimale est $\{b, c, d\}$. A présent, considérons (G, U) , où $U = \{a, b\}$, comme une instance de EXT_{\min} IS. Supposons que S est un stable maximal de G . Si $a \in S$, alors par maximalité $e \in S$, et si $b \in S$, alors $\{c, d\} \subset S$. De manière similaire au problème EXT_{\min} IS, nous cherchons un stable maximal qui inclut un nombre minimal de sommets en dehors de U , alors la solution optimale de EXT_{\min} IS pour (G, U) est $\{a, e\}$.

Pour $S = \text{EXT}_{\max}\Pi$ ou $\text{EXT}_{\min}\Pi$, on note $\text{opt}_S(x, U)$ la valeur de la solution optimale. Notons que dans les deux problèmes mentionnés $\text{opt}_S(x, U) = |U|$ si et seulement si (x, U) est une instance postive de EXT Π . Ainsi, on peut conclure qu'un problème anti-héréditaires (resp. héréditaires) Π , $\text{EXT}_{\max}\Pi$ (resp. $\text{EXT}_{\min}\Pi$) est NP-difficile dès lors que EXT Π est NP-complet.

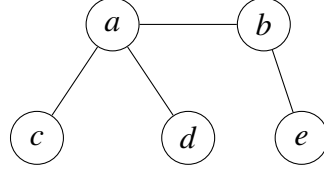


Figure 3: Le graphe $G = (V, E)$ comme instance pour $\text{EXT}_{\max}\text{VC}$ et $\text{EXT}_{\max}\text{IS}$.

Il est aisé de voir que les solutions optimales de $\text{EXT}_{\max}\text{VC}$ et $\text{EXT}_{\min}\text{IS}$ sont liés par une relation analogue à celle existante entre $\alpha(G)$ et $\tau(G)$ (i.e. le nombre de sommets dans la couverture par sommets). Par conséquent, pour tout graphe $G = (V, E)$ ayant n sommets, nous avons $\text{opt}_{\text{EXT}_{\max}\text{VC}}(G, U) + \text{opt}_{\text{EXT}_{\min}\text{IS}}(G, V \setminus U) = n$.

Le prix de l'extension PoE est défini comme le ratio d'approximation, i.e. $\frac{\text{apx}}{\text{opt}}$. Nous stipulons que Π admet un ρ - PoE polynomial si pour toute instance (x, U) , on peut trouver une solution S de G en temps polynomial qui satisfait $\text{PoE}(S) \geq \rho$ pour $\text{EXT}_{\max}\Pi$ (resp. $\text{PoE}(S) \leq \rho$ pour $\text{EXT}_{\min}\Pi$). Ceci est présenté formellement dans la définition ci-après :

Definition 0.4. Pour une instance (x, U) de $\text{EXT}_{\max}\Pi$ et (resp. $\text{EXT}_{\min}\Pi$), le prix de l'extension (PoE) d'une solution minimale (resp. maximale) S de Π sur x est défini par $\text{PoE}(S) := \frac{|S \cap U|}{\text{opt}_{\text{EXT}_{\max}\Pi}(x, U)}$ (resp. $\text{PoE}(S) := \frac{|S \cup U|}{\text{opt}_{\text{EXT}_{\min}\Pi}(x, U)}$).

Dans la section qui suit, nous introduirons un cadre de travail de *maximum-minimal* (max-min) et *minimum-maximal* (min-max) et nous montrons la relation qui peut exister entre $\text{EXT}_{\max}\Pi$ et $\text{EXT}_{\min}\Pi$ dans ce contexte.

Problèmes d'optimisation max-min et min-max dans les graphes

Dans cette section, nous abordons des problèmes d'optimisation dits *max-min*, *min-max* dans les graphes issus de la thèse de Manlove [136]. Par la suite, nous décrivons la relation entre les problèmes d'extension et d'optimisation min-max ou max-min. Nous terminons cette section par un passage en revue des principaux résultats liés aux problèmes d'optimisation max-min ou min-max.

Nous présentons dans cette partie la manière qui nous permet d'obtenir un problème d'optimisation Π' de *min-max* à partir d'un problème source Π .

Definition 0.5. Soit $\Pi = \langle I, F, \text{opt}, m \rangle$ un problème d'optimisation héréditaire ou anti-héréditaire. Alors le nouveau problème d'optimisation est $\Pi' = \langle I, F', \text{opt}', m \rangle$, où pour chaque instance $G \in I$:

- $F'(G) := \begin{cases} \mu(G) & \text{si } \Pi \text{ est un problème anti-héréditaire} \\ \psi(G) & \text{si } \Pi \text{ est un problème héréditaire} \end{cases}$
- $\text{opt}' := \begin{cases} \min & \text{si } \text{opt} = \max \\ \max & \text{si } \text{opt} = \min \end{cases}$

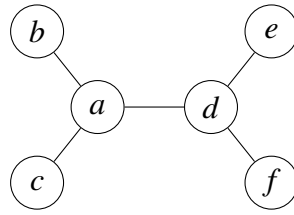


Figure 4: Le graphe G comme instance de MAX-MIN VC and MIN-MAX IS.

Si $opt = max$ alors le nouveau problème Π' est un problème *min-max* appelé aussi *lower* Π et si $opt = min$, Π' est un problème *max-min* appelé aussi *upper* Π dans le cadre proposé par Manlove dans [136]. L'auteur a étudié de nombreux problèmes de manière systématique, en considérant différents ordres partiels(voir [137, 112] pour la coloration et 12 pour les problèmes de couverture et d'indépendance dans les graphes. Comme exemples d'ordres partiels qui ne sont pas héréditaires ou anti-héréditaires le problème de k -échange pour un stable[135] ou les problèmes appelés Somme de sous-ensemble maximale avec contraintes digraphique [97]. Une série d'autres problèmes qui ne sont ni héréditaires ni anti-héréditaires peuvent être trouvés dans [136]. Ceux-là sont discutés dans le Chapitre 6. Par exemple, si Π est MIN VC (resp. MAX IS), alors Π' s'intéresse à trouver une couverture par sommet minimale (resp. stable maximal) de taille maximum (resp. minimum). Les deux nouveaux problèmes ont été formellement définis comme suit:

MAX-MIN VC (aussi appelé UPPER VC)

Entrée: Un graphe $G = (V, E)$.

Solution: une couverture par sommet minimale $S \subseteq V$.

Sortie: $|S|$ maximisé.

MIN-MAX IS (aussi appelé LOWER IS)

Entrée: Un graphe $G = (V, E)$.

Solution: Un stable maximal $S \subseteq V$.

Sortie: $|S|$ minimisé.

Considérons l'exemple suivant pour plus de clarté.

Example 0.6. Soit le graphe de la Figure 4 qu'on considère comme une instance des deux problèmes MAX-MIN VC et MIN-MAX IS. Il y a 3 couvertures par sommets minimales différentes de G , $\{a, e, f\}$, $\{a, d\}$ et $\{d, b, c\}$ alors que dans le problème MAX-MIN VC, on cherche une couverture par sommet minimale de taille maximum, alors la taille d'une solution optimale est 3 dans ce cas précis. En ce qui concerne le MIN-MAX IS il existe aussi 3 solutions réalisables pour G , $\{b, c, d\}$, $\{a, e, f\}$ et $\{b, c, e, f\}$, car on cherche, pour le problème de MIN-MAX IS, un stable maximum de taille minimale, donc la taille d'une solution optimale est de 3.

Un cas particulier de $Ext_{min}\Pi$ (resp. $Ext_{max}\Pi$) pour une instance (x, U) se produit quand la solution partielle U est l'ensemble vide (resp. l'ensemble entier). Ainsi, si Π est un problème

anti-héréditaire, tel $Ext_{max}\Pi$ ayant (x, U) comme instance, alors celui-ci est exactement équivalent au fameux problème MAX-MIN Π ou bien au UPPER Π quand U est l'ensemble entier. De plus, si Π est un problème héréditaire, comme $Ext_{min}\Pi$ avec (x, U) comme instance, celui-ci est exactement équivalent à MIN-MAX Π ou LOWER Π quand U est l'ensemble entier. Par conséquent, $Ext_{max}\Pi$ contient le MAX-MIN Π pour un problème anti-héréditaire $Ext_{min}\Pi$ contient le MIN-MAX Π pour un problème héréditaire Π .

Supposons que $G = (V, E)$ est une instance de deux problèmes MIN VC et MAX IS. Le but de $Ext_{max_{max}}$ VC, dans le cas où la solution partielle $U = V$, est de trouver une couverture par sommet minimale contenant le nombre maximum de sommets de V , qui est une solution pour MAX-MIN VC, tandis que le but de Ext_{min} IS, quand $U = \emptyset$, est de trouver une couverture par sommet maximale contenant le nombre minimum de sommets de V qui est une solution pour MIN-MAX IS.

Considérer une version max-min ou min-max d'un problème en "retournant" l'objectif n'est pas une idée nouvelle; en effet, une telle question a déjà été abordée pour de nombreux problèmes classiques d'optimisation. Certains des exemples les plus connus incluent le problème MIN-MAX IS [34]. Le problème MIN-MAXVC [30, 163], le problème LAZY BUREAUCRAT qui est une version min-max du problème SOMME DE SOUS-ENSEMBLE qui cherche à savoir si pour un ensemble d'entiers w_1, \dots, w_n et une valeur W , il existe un sous-ensemble dont la somme des éléments est exactement W ? [86, 96], le problème MAX-MIN DS [2, 17], et le problème MIN-MAX MATCHING [162].

Le problème MIN-MAX IS, aussi appelé *l'ensemble dominant et stable minimum (minimum independent dominating set)* (MIN ISDS) cherche, étant donné un graphe $G = (V, E)$, un sous-ensemble $S \subseteq V$ de taille minimale qui est simultanément indépendant et dominant. Du point de vue complexité, la difficulté de la résolution exacte de MIN ISDS est équivalente à UPPER VC [137]. Toutefois, ces deux problèmes semblent avoir des comportements qui divergent en termes d'approximation et de complexité paramétrée [12]. Bien que MIN ISDS soit polynomial dans les graphes cordaux [77], il est difficile de se rapprocher de $n^{(\epsilon-1)}$, pour tout $\epsilon > 0$, dans certaines classes de graphes [77, ?]. En ce qui concerne la complexité paramétrée, Fernau [81] a présenté un algorithme FPT pour UPPER VC avec un temps d'exécution $O^*(2^k)$, où k est la taille d'une solution optimale, alors qu'il a été prouvé que MIN ISDS est $W[2]$ -difficile pour le paramètre standard. En outre, Boria et al. [30] ont fourni un fort résultat d'approximation pour UPPER VC dans les graphes simples: les auteurs ont, non seulement, décrit un algorithme d'approximation avec un ratio de $n^{\frac{1}{2}}$ mais aussi prouvé qu'il est NP-difficile d'approximer que le problème UPPER VC à $n^{(\epsilon-\frac{1}{2})}$, pour tout $\epsilon > 0$. De plus, ils ont présenté un algorithme paramétré avec un temps d'exécution (1.5397^k) où k est le paramètre standard et en modifiant l'algorithme proposé par [81]; les auteurs ont aussi montré que les versions pondérées de UPPER VC et MIN ISDS sont FPT par rapport à la largeur arborescente.

Le problème MIN-MAX MATCHING a été étudié dès 1980, lorsque Yannakakis et Gavril ont prouvé qu'il est NP-difficile même dans des cas très restreints [162]. MIN-MAX MATCHING est équivalent au *minimum edge dominating set (l'ensemble d'arêtes dominant minimum)* (dénote par MIN EDS) ; le problème s'intéresse à déterminer si, étant donné un graphe $G = (V, E)$, pour

un sous-ensemble $S \subseteq E$ qui domine toutes les arêtes du graphe, cet ensemble S est de taille minimale. Un ensemble dominant et stable d'arêtes est un ensemble dominant d'arêtes dans lequel il n'existe pas deux éléments adjacents. Le problème MIN-MAX MATCHING, aussi connu sous le nom de *minimum independent edge dominating set* et noté par MIN ISEDS) s'intéresse à trouver si, étant donné un graphe $G = (V, E)$, le sous-ensemble $S \subseteq E$ qui est simultanément indépendant et dominant est de taille minimale. En 2006, il a été prouvé qu'il est NP-difficile d'approximer le MIN-MAX MATCHING par un meilleur facteur que $\frac{7}{6}$ [50]. Plus tard, dans [75], les auteurs proposent un résultat amélioré qui prouve que le problème est difficile à approximer avec un ratio supérieur à 1.18, en supposant que $P \neq NP$. Enfin, Dudycz et al. ont prouvé que le MIN-MAX MATCHING est difficile à approximer avec un ratio de 2, en supposant la conjecture des jeux unique (Unique Games Conjecture (UGC)) [73].

Aperçu

Dans cette thèse, nous étudions un type particulier de problèmes d'optimisation, appelés *problèmes d'extension*. Nous étudions certains problèmes d'optimisation héréditaires et anti-héréditaires dans les graphes du point de vue de la complexité, de l'(in-)approximation et de la complexité paramétrée. De manière générale, quand il s'agit de la version d'extension d'un problème d'optimisation anti-héréditaire (ou héréditaire) Π , nous considérons une instance de Π avec un ensemble supplémentaire (spécifié) U appelé solution partielle, le but étant d'obtenir une solution S *minimal* pour l'inclusion (resp. maximal) et de valeur optimale telle que $U \subseteq S$ (resp. $U \supseteq S$).

Concernant la recherche d'une solution minimale contenant un ensemble forcé, avec une valeur optimale, nous étudions le problème pondéré de la couverture par arêtes dans le Chapitre 3. Pour un graphe donné $G = (V, E)$, l'ensemble $S \subseteq E$ est une couverture par arêtes si et seulement si S couvre tous les sommets de V , c.-à-d. chaque sommet $v \in V$ est incident à au moins un $e \in S$. Le but de la version d'optimisation du problème est de trouver une couverture par arête de taille minimale. La variante pondérée de la couverture par arêtes prend $G = (V, E, w)$ un graphe et une pondération sur les arêtes en entrée, le but étant de trouver une couverture par arête S avec une valeur optimale, i.e. en minimisant $w(S) = \sum_{e \in S} w(e)$. Dans ce chapitre, nous étudions la version pondérée de la couverture par arêtes dans le cadre des problèmes d'extension.

L'entrée de la version d'extension de la couverture par arêtes pondérée comprend à la fois un graphe pondéré sur les arêtes $G = (V, E, w)$ ainsi qu'un ensemble forcé $U \subseteq E$, la tâche est d'étendre U à une couverture par arête minimale de valeur optimale, du point de vue inclusion. Les variantes de maximisation et de minimisation du problème sont étudiées dans ce chapitre. Nous étudions les deux problèmes du point de vue complexité et de l'(in-)approximation.

En premier lieu, nous montrons que pour un graphe donné G et un ensemble forcé U , EXT-EC est NP-complet même pour les graphes bipartis de degré maximum 3. Ensuite, nous considérons les variantes de maximisation et de minimisation du problème dans les graphes complets. Pour

une fonction de poids générale, nous montrons que **MIN EXT WEC** n'est pas approximable, mais **MAX EXT WEC** est approximable avec un ratio de 0,5. Par suite, nous considérons d'autres fonctions de poids et nous montrons que **MIN EXT WEC** est dans **APX** lorsque la fonction de poids satisfait l'inégalité triangulaire c -relaxée (ou c -relaxée et étendue) pour $c > 1$. De plus, le problème devient polynomial lorsque la fonction de poids satisfait l'inégalité triangulaire du c -relaxée (resp. c -relaxée étendue) et que $c \leq 1$ (resp. $c = 1$). En outre, nous démontrons que les deux problèmes d'optimisation peuvent être résolus en un temps polynomial dans les graphes à largeur arborescente bornée. La Table 1 résume les résultats de complexité des deux problèmes d'optimisation dans les graphes complets.

| | <i>w-générale</i> | <i>c-relaxée</i> | <i>c-relaxée et étendue</i> |
|--------------------|--|--|---|
| MIN EXT WEC | NP-difficile non-apx FPT p.r U | NP-difficile $c > 1$ polynomial $\frac{1}{2} \leq c \leq 1$ c ratio-apx $\frac{7+c}{8} - \epsilon$ in-apx | NP-difficile $c > 1$ polynomial $c = 1$ $\frac{c+1}{2}$ ratio-apx $\frac{7+c}{8} - \epsilon$ non-apx |
| MAX EXT WEC | 0.5 ratio-apx $\frac{7}{8} + \epsilon$ in-apx | — | — |

Table 1: Résultats pour les graphes complets

Nous nous concentrons sur un cas particulier de **MAX EXT WEC**, **WEIGHTED UPPER EC**, lorsque l'ensemble forcé est vide. Etant donné un graphe avec arêtes pondérées, l'objectif est de trouver une couverture par arêtes minimale ayant un poids total maximum. Nous étudions ce problème dans certaines classes de graphes d'un point de vue de l'(in)approximation : nous montrons que **WEIGHTED UPPER EC** dans les graphes complets est équivalent au problème de **MAX STAR FOREST** dans les graphes simples. Nous montrons également que, **WEIGHTED UPPER EC** dans les graphes bipartis et les split graph ayant pour poids 0 ou 1 est aussi difficile que **MAX IS** dans les graphes simples. De plus, nous montrons que ce problème est **APX** complet dans les k -arbres et dans les graphes de degré borné.

Nous considérons les variantes d'extension de certains problèmes de sommets dans le Chapitre 4. Nous étudions le problème de décision **EXT Π** lorsque Π fait partie d'un des problèmes de sommets suivants : **VERTEX COVER** (COUVERTURE PAR SOMMETS), **INDEPENDENT SET** (STABLE), **CONNECTED VERTEX COVER** (COUVERTURE PAR SOMMETS CONNEXE), **NON SEPARATING INDEPENDENT SET** (STABLE NON-SÉPARANT) et **DOMINATING SET** (ENSEMBLE DOMINANT). Un sous-ensemble $S \subseteq V$ d'un graphe connexe donné $G = (V, E)$ est une couverture par sommets connexe (Connected Vertex Cover) (resp. *un stable non-séparant*) si S est une couverture par sommets et le graphe $G[S]$ induit par S est connexe (resp. si S est un stable et n'est pas une coupe de G). Le but de la variante d'optimisation des deux problèmes précédents est de trouver une couverture par sommets connexe (resp. un stable non séparant) de taille minimale (resp. maximale). En d'autres termes, pour un problème de sommets héréditaire (resp. anti-héréditaire) Π , le problème de décision noté **EXT Π** , dans lequel une paire (G, U) est donnée en entrée telle que $G = (V, E)$ et

$U \subseteq V$, celui-là détermine s'il existe une solution S de sorte que $S \subseteq U$ (resp. $U \subseteq S$) avec un poids maximum (resp. minimum).

En ce qui concerne ces problèmes de décision, nous obtenons des résultats de complexité même pour des instances restreintes à des graphes bipartis ou planaires. La plupart de nos résultats de NP-difficulté se traduisent également en des résultats ETH-difficile. Le Table 2 résume les résultats de complexité obtenus pour ces problèmes.

| | NP-complet | Polynomial |
|---------------------|--|---|
| EXT VC/IS | graphes bipartis sous-cubiques et planaires graphes bipartis cubiques | graphe cordaux graphes avec arcs circulaires |
| EXT CVC/NSIS | graphes bipartis sous-cubiques graphes faiblement triangulaires | graphes cordaux |
| EXT DS | graphes bipartis sous-cubiques et planaires | |

Table 2: Résultats de complexité

De plus, nous étudions la complexité paramétrée de ces problèmes, que ce soit par rapport à la taille de la solution partielle U ou celle de l'ensemble dual \bar{U} . Nous étudions l'impact de ces paramètres dans des scénarios et classes restreints, qu'il s'agisse de degrés ou de restrictions topologiques (graphes bipartis, planaires ou cordaux). Le Table 3 résume les résultats paramétrés obtenus pour ces problèmes.

| | EXT VC | EXT IS | EXT CVC | EXT NSIS | EXT DS |
|---------------------------|---------------|---------------|----------------|-----------------|---------------|
| paramètre standard | W[1]-com | FPT | W[1]-com | FPT | W[3]-com |
| paramètre dual | FPT | W[1]-com | FPT | W[1]-com | FPT |

Table 3: résultats de complexité paramétrée

D'autre part, nous étudions le prix de l'extension (PoE), une mesure qui reflète la distance d'un ensemble de sommets U d'un de sous-ensemble maximum qui peut être étendu à une solution minimale/maximale, et qui fournit des résultats négatifs et positifs du point de vue approximation pour la PoE dans différentes classes de graphes.

Nous introduisons la notion de *prix de l'extension* (PoE), qui permet de quantifier la distance de U de l'ensemble le plus proche qui peut être étendu pour obtenir une solution. Ceci résulte

naturellement en l'apparition de problèmes d'optimisation liés aux problèmes d'extension pour lesquels nous fournissons des résultats d'approximation. De plus, pour certains problèmes tels la COUVERTURE PAR SOMMETS CONNECTÉS, nous nous concentrons sur un cas particulier (G, \emptyset) de $\text{EXT}_{\max}\text{CVC}$, qui est aussi connu sous le nom de MAX-MIN CVC. Nous étudions ce dernier problème d'un point de vue de l'(in)approximation et présentons quelques résultats intéressants.

Dans le Chapitre 5, nous abordons en détails les extensions de problèmes d'arêtes dans les graphes. Plus précisément, nous considérons *les problèmes d'extension* lié à plusieurs problèmes d'optimisation d'arêtes classiques dans les graphes, à savoir EDGE COVER, MATCHING et EDGE DOMINATING SET. Étant donné un graphe $G = (V, E)$ et un ensemble d'arêtes $U \subseteq E$, on essaie de déterminer s'il existe une solution minimale (en termes d'inclusion) (resp. maximale) réalisable E' qui satisfait une propriété donnée, par exemple, étant donné un ensemble dominant (resp. un matching d'arêtes) et contenant un *ensemble forcé* (resp. ou être inclus dans le *ensemble co-interdit*) U .

Soit $G = (V, E)$ un graphe où le degré minimum est d'au moins $r \geq 1$. Pour une constante fixe r :

- $S \subseteq E$ est appelé une couverture de G , si chaque sommet $v \in V$ est incident à au moins r arêtes de S . Le cas spécial $r = 1$ est connu sous le nom de *couverture par arêtes* (*Edge cover*).
- $S \subseteq E$ est appelé un ensemble dominant de r -arêtes si chaque arête $e \in E$ est incidente à au moins r arêtes de S . Le cas spécial $r = 1$ est connu le nom d'*ensemble dominant* (*Dominating Set*).
- $S \subseteq E$ est appelé un sous-graphe partiel à contrainte de degré r si aucun des sommets en V n'est incident à plus de r arêtes en S . Le cas spécial $r = 1$ est connu sous le nom de *couplage*.

De fait, nous considérons les variantes d'extension des problèmes d'arêtes ci-dessus dans leur version généralisée, lorsque la r -contrainte est incluse dans la définition du problème et ne figure pas dans la description de l'entrée de l'instance. Par exemple, pour un graphe donné $G = (V, E)$ avec un degré minimum d'au moins r et un ensemble d'arêtes forcées $U \subseteq E$ comme instance de la version d'extension de la couverture par r arêtes dénotée $\text{EXT } r\text{-EC}$, le but est de déterminer s'il existe une couverture minimale (pour l'inclusion) de r arêtes E' contenant l'ensemble d'arêtes *forcées* U .

Nous présentons différents résultats de complexité pour ces problèmes, et nous prouvons que ces problèmes restent NP-complets, même dans les graphes bipartis de degré limité avec certaines contraintes imposées sur l'ensemble des arêtes forcées/co-interdites. La planarité ne diminue en rien la difficulté du problème, nous montrons effectivement que ces problèmes restent difficiles sur les graphes planaires bipartis sous-cubiques. En contrepartie, nous fournissons des résultats de complexité paramétrée qui contrebalancent les résultats négatifs développés, il est à noter que nous considérons la taille de l'ensemble forcé des arêtes (resp. ensemble co-interdit) comme paramètre standard pour la variante d'extension de EDGE COVER et de EDGE DOMINATING SET. (resp. MATCHING). Nous considérons également le prix de l'extension (PoE), qui est une

variante naturelle des problèmes d'optimisation d'extension et qui conduit à certains résultats d'approximation.

De même, nous montrons que $\text{EXT } r\text{-EC}$, $\text{EXT } r\text{-EDS}$ et $\text{EXT } r\text{-DCPS}$ sont NP-complets pour les graphes de degré maximum $r + 2$, et nous prouvons que pour le cas particulier où $r = 1$, tous les problèmes mentionnés sont NP-complet pour les instances de graphes bipartis planaires sous-cubiques. Nous étudions également la complexité paramétrée de ces problèmes, par rapport à la taille de la solution partielle U ou de son dual \bar{U} , qui conduit à des résultats de complexité paramétrée positifs. Le Table 4 résume les résultats de complexité et les résultats de complexité paramétrée obtenus pour ces problèmes.

| | NP-complétude | | Complexité Paramétrée |
|---------------------------------------|--|--|--------------------------|
| | $r \geq 2$ | $r = 1$ | |
| EXT $r\text{-EC}$ | graphes bipartis avec $\Delta = r + 2$ | graphes bipartis planaires sous-cubiques | FPT par $ U $ |
| EXT $r\text{-EDS}$ | graphes bipartis planaires avec $\Delta = r + 2$ | graphes bipartis planaires sous-cubiques | W[1]-difficile par $ U $ |
| EXT $r\text{-DCPS}$ | graphes bipartis avec $\Delta = r + 1$ | graphes bipartis planaires sous-cubiques | FPT par $ \bar{U} $ |

Table 4: Résultats de complexité et de complexité paramétrée.

Nous avons porté une attention particulière à la discussion du *prix de l'extension* (PoE), qui est une mesure de la distance entre l'ensemble U et l'ensemble le plus proche qui peut être étendu. Ceci génère de nouveaux problèmes d'optimisation liés aux problèmes d'extension pour lesquels nous développons des résultats d'approximation.

Les résultats que nous exposons dans le Chapitre 5 vont être particulièrement utiles pour la communauté des algorithmes d'énumération (sensibles par rapport à l'entrée) qui n'a jusqu'à présent pas accordé une attention particulière aux problèmes sur les arêtes des graphes; les seuls travaux sur ce sujet étant [95]. À l'inverse, les algorithmes d'énumération sensibles à la sortie, comme pour les correspondances, existent depuis plus de vingt ans [157]. Quelques réflexions sur l'énumération de la couverture par arêtes peuvent être trouvées dans [160].

Enfin, dans le Chapitre 6, nous introduisons un cadre général d'étude basé sur l'ordre partiel pour modéliser l'extension de problèmes monotones. Nous y soulignerons les nombreuses similarités identifiées entre ces problèmes dans des scénarios divers. La question "une solution partielle donnée à un problème peut-elle être étendue ?" se pose dans de nombreuses approches algorithmiques pour les problèmes d'optimisation. Dans le Chapitre 4 et le Chapitre 5, nous avons étudié quelques problèmes d'optimisation de sommets et d'arêtes dans les graphes. Cependant, il existe un grand nombre de problèmes d'optimisation NP tels que les problèmes de chaînes de caractères, les problèmes de logique, de coloration de graphe, etc., dont les variantes d'extension

semblent intéressantes. Il est vrai qu'une abondance de ces problèmes ont été considérés dans le cadre des problèmes d'extension [23, 109, 84, 7, 8, 113, 139, 109, 146, 14, 159, 138, 23], toutefois, notre approche est fondamentalement différente dans le sens où nous imposons des conditions de minimalité.

Le sujet principal du Chapitre 6 est de proposer un cadre général pour ce type de problèmes d'extension. L'idée derrière est basée sur une approche d'ordre partiel, rappelant ce qui a été fait dans le cadre de cette thématique [136]. Essentiellement, nous considérons les problèmes d'optimisation dans NPO avec un ensemble de solutions partielles spécifiées que nous appelons *solutions partielles* (y compris l'ensemble des solutions) et un ordre partiel sur celles-ci. Cet ordre partiel \leq reflète non seulement la notion d'*extension* mais aussi de *minimalité*. Pour une solution partielle U et une solution S , S étend U si $U \leq S$. Une solution S est *minimale* s'il n'existe pas de solution $S' \neq S$ avec $S' \leq S$. Le problème d'extension qui en résulte est défini comme étant le problème de décision que si nous disposons d'une pré-solution donnée U , il existe une solution minimale S qui étend U . Nous ajoutons également des résultats de complexité pour des problèmes tels que BIN PACKING, FEEDBACK VERTEX SET et FEEDBACK EDGE SET.

Bien que l'étude que nous avons menée ne couvre pas la totalité des problèmes qui puissent être reliée au sujet en questions, des problèmes similaires ont été examinés, par exemple, dans le domaine de la coloration de graphes, sous le nom de *pre-coloring extension*. Ce dernier contient l'étude de la complétion des carrés latins partiels comme un cas particulier [23, 54, 138]. Cependant, il y a une différence majeure avec notre approche qui réside dans le fait qu'à l'instar des problèmes que nous avons évoqués où la condition de minimalité sur les extensions autorisées est essentielle, elle devient néanmoins pourvue de son intérêt dans le meilleur des cas pour les problèmes d'extension de pré-coloration. Il n'est ainsi pas difficile de définir un ordre partiel sur les pré-colorations pour garantir que l'ensemble des couleurs propres soit fermé comme requis dans le contexte de nos travaux. Il serait alors intéressant de considérer ce type de problèmes d'extension dans un contexte plus approfondi permettant d'obtenir des résultats généraux, des méta-théorèmes, etc.

Contents

| | |
|---|-----------|
| 1 Preliminaries | 29 |
| 1.1 Graphs | 29 |
| 1.1.1 Basics of graphs | 29 |
| 1.1.2 Some graph classes | 31 |
| 1.1.3 Tree decomposition | 32 |
| 1.2 Computational complexity | 33 |
| 1.2.1 NP-hardness and polynomial solvability | 34 |
| 1.2.2 Approximation of NP optimization problems | 37 |
| 1.2.3 Parameterized complexity | 39 |
| 2 Extension of Minimal and Maximal Graph Problems | 47 |
| 2.1 Extension optimization problems | 49 |
| 2.2 Decision problems associated with extension optimization problems | 50 |
| 2.3 Price of extension | 53 |
| 2.4 Max-min and min-max optimization problems in graphs | 54 |
| 2.5 Overview | 57 |
| 3 Extended Weighted Edge Cover Problem | 61 |
| 3.1 Introduction | 62 |
| 3.1.1 Some general definitions | 64 |
| 3.1.2 Related work | 65 |
| 3.1.3 Contributions | 67 |
| 3.2 Extended weighted edge cover in general graphs | 67 |
| 3.3 MIN EXT WEC in complete graphs | 73 |
| 3.4 MAX EXT WEC in complete graphs | 81 |
| 3.5 Weighted upper edge cover | 84 |
| 3.5.1 Weighted Upper EC in complete graphs | 85 |

| | | |
|----------|--|------------|
| 3.5.2 | Weighted Upper EC in bipartite graphs | 87 |
| 3.5.3 | Weighted Upper EC in split graphs | 89 |
| 3.5.4 | Weighted Upper EC in k -trees | 91 |
| 3.5.5 | Weighted Upper EC in bounded degree graphs | 97 |
| 3.6 | Conclusion | 98 |
| 4 | Extension of Some Vertex Graph Problems | 101 |
| 4.1 | Introduction | 102 |
| 4.1.1 | Related work | 102 |
| 4.1.2 | Contributions | 104 |
| 4.2 | EXT VC, EXT IS | 104 |
| 4.2.1 | Computational complexity | 106 |
| 4.2.2 | Parameterized complexity | 113 |
| 4.2.3 | Price of extension | 115 |
| 4.2.4 | Generalizations to extensions of H -graph cover and H -free subgraph | 118 |
| 4.3 | EXT CVC and EXT NSIS | 120 |
| 4.3.1 | Computational complexity | 121 |
| 4.3.2 | Price of extension | 126 |
| 4.3.3 | Upper CVC | 130 |
| 4.4 | EXT DS | 133 |
| 4.4.1 | Computational complexity | 133 |
| 4.4.2 | Parameterized complexity | 137 |
| 4.4.3 | Price of extension | 138 |
| 4.5 | Conclusion | 139 |
| 5 | Extension of Some Edge Graph Problems | 141 |
| 5.1 | Introduction | 142 |
| 5.1.1 | Related work | 142 |
| 5.1.2 | Contributions | 143 |
| 5.2 | Ext EC and its generalization | 143 |
| 5.2.1 | Computational complexity | 144 |
| 5.2.2 | Parameterized complexity | 148 |
| 5.2.3 | Price of extension | 149 |
| 5.3 | Ext EDS and its generalization | 150 |
| 5.3.1 | Computational complexity | 151 |
| 5.3.2 | Parameterized complexity | 152 |
| 5.3.3 | Price of extension | 154 |
| 5.4 | Ext EM and its generalization | 156 |
| 5.4.1 | Computational complexity | 156 |
| 5.4.2 | Parameterized complexity | 162 |
| 5.4.3 | Price of extension | 164 |
| 5.5 | Conclusion | 165 |

| | | |
|----------|---|------------|
| 6 | General Framework of Extension Problems | 167 |
| 6.1 | A General framework of extension problems | 168 |
| 6.2 | Some partial orders | 170 |
| 6.3 | Some new extension problems | 171 |
| 6.4 | Conclusion | 176 |
| 7 | Conclusion and Future Work | 177 |

Contents

| | | |
|------------|---|-----------|
| 1.1 | Graphs | 29 |
| 1.1.1 | Basics of graphs | 29 |
| 1.1.2 | Some graph classes | 31 |
| 1.1.3 | Tree decomposition | 32 |
| 1.2 | Computational complexity | 33 |
| 1.2.1 | NP-hardness and polynomial solvability | 34 |
| 1.2.2 | Approximation of NP optimization problems | 37 |
| 1.2.3 | Parameterized complexity | 39 |

In this chapter, we give some basic notions of graph theory as well as some basic background on optimization problems, computational complexity, (in-)approximability and parameterized complexity.

1.1 Graphs

1.1.1 Basics of graphs

A graph is a pair of sets (V, E) where E is a set of unordered pairs of V . Throughout this thesis, we consider a simple graph¹ $G = (V, E)$ where V is called *vertex set* and E is called *edge set*. Sometimes we assume that G is an edge-weighted (resp. a vertex-weighted) graph. In this case, it is denoted by a tuple $G = (V, E, w)$ where w is a set of non-negative values on the edge set E (resp. the vertex set V) of G . For simplicity, we denote every edge $\{u, v\} \in E$ by uv for any pair of vertices $u, v \in V$. The graph $G = (V, E)$ is *directed* if each edge $e \in E$ is ordered. We denote by (u, v) a directed edge from vertex u to vertex v . The *complement* of the graph $G = (V, E)$, is

¹without multiple edges and loops.

denoted by $\overline{G} = (V, \overline{E})$ in which $uv \in E$ if and only if $uv \notin \overline{E}$ for all pairs of vertices $u, v \in V$, $u \neq v$.

Every edge $e = uv$ has two endpoints u and v . If v is an endpoint of the edge e , we say that e and v are *incident*. Let $G = (V, E)$ be a graph and $U \subseteq V$, $N_G(U) = \{v \in V : vu \in E, \text{ for some } u \in U\}$ denotes *the neighborhood* of U in G and $N_G[U] = U \cup N_G(U)$ denotes *the closed neighborhood* of U . For singleton sets $U = \{u\}$, we simply write $N_G(u)$ or $N_G[u]$, even omitting G if it is clear from the context. The cardinality of $N_G(u)$ is called *degree* of u , and denoted by $d_G(u)$ and for a subset $U \subseteq V$, $d_U(v)$ is the number of neighbors of v belonging to U . A graph $G = (V, E)$ is said to be of *maximum* (resp. *minimum*) *degree* p , if and only if for any vertex $v \in V$, $d_G(v) \leq p$ (resp. $d_G(v) \geq p$). If all vertices have degree p , it is called *p-regular*. *3-regular* graphs are called also *cubic* and if 3 upper-bounds the degree of all vertices we speak of *subcubic graphs*. The *maximum degree* of the graph G is denoted $\Delta(G) = \max_{v \in V} d_G(v)$. A *leaf* is a vertex of degree one, it is also called *pendant* vertex and the edge incident to a pendant vertex is called *pendant edge*.

A *subgraph* of G is a graph $H = (V_H, E_H)$ such that $V_H \subseteq V$ and $E_H \subseteq E$ where $uv \in E_H$ implies that $u, v \in V_H$. It is called *induced by vertices* V_H and denoted by $G[V_H]$, if for each pair of vertices $u, v \in V_H$, $uv \in E_H$ if and only if $uv \in E$.

For a given graph $G = (V, E)$ and a subset $E' \subseteq E$, the set of endpoints of E' , is denoted by $V(E')$, i.e. $V(E') = \{u, v : uv \in E'\}$. The subgraph induced by edges E' , denoted $G[E']$, is defined by $G[E'] = (V(E'), E')$, when E' is full that is $uv \in E \setminus E'$ implies $u \notin V(E')$ or $v \notin V(E')$; in this case, we have $G[E'] = G[V(E')]$. Finally, the partial graph induced by E' denoted $G_{E'}$ is defined by $G_{E'} = (V, E')$.

For a given graph $G = (V, E)$, a *path* is a sequence of distinct vertices (v_1, \dots, v_k) of V such that $v_i v_{i+1} \in E$ for $1 \leq i \leq k-1$ and a *cycle* is a path v_1, \dots, v_k of V such that $v_1 = v_k$. The *length* of a path (resp. cycle) involving a sequence of k vertices, is $k-1$ (resp. k) and such a path (resp. cycle) is denoted by P_k (resp. C_k).

A graph G is *connected* if for each pair of vertices u, v , there exists a path between u and v . A graph is *k-connected* if by removing any subset of $k-1$ vertices of the graph, it remains connected. A *2-connected* graph, is also called *bi-connected*. A *connected component* (resp. *k-connected component*) of a graph G is an inclusion-wise maximal connected (resp. *k-connected*) induced subgraph of G .

A graph $G = (V, E)$ with $|V| = n$ is called *c-dense* if it contains at least $c \frac{n^2}{2}$ edges and it is called *everywhere-c-dense* if the minimum degree is cn ; when $c \in (0; 1)$ is a constant, we say *dense* and *everywhere-dense* graphs.

A *cut-set* (also called *vertex separator*) is a subset of vertices such that deleting them from G strictly increases the number of connected components. A cut-set which is a singleton is called a *cut-vertex* (or *articulation point*). Hence, a graph is bi-connected if and only if it is connected and it does not contain any cut-vertex.

A partition of a vertex set V is a set of subsets $\{V_1, \dots, V_k\}$ of V for some integer k such that $\cup_{i=1}^k V_i = V$ and $V_i \cap V_j = \emptyset$ for any pair $i, j \in \{1, \dots, k\}$ and $i \neq j$. A *vertex-coloring* (or *coloring*)

of a graph $G = (V, E)$ is an assignment of colors to V . A coloring f is *proper*, if $f(u) = f(v)$ for any two vertices $u, v \in V$, then $uv \notin E$. A proper coloring which uses at most k colors is called *k-coloring*. The smallest number of colors to make a proper coloring of G is called *chromatic number* and denoted by $\chi(G)$.

For a graph $G = (V, E)$, a subset $C \subseteq V$ is a *clique* if for any two vertices $u, v \in C$, $uv \in E$; a subset $I \subseteq V$ is an *independent set* if for any two vertices $u, v \in I$, $uv \notin E$; a subset $M \subseteq E$ is a *matching* if any two edges of M are pairwise non-adjacent; a matching M of G is *perfect* if M is incident to all vertices of V ; a *claw* is a $K_{1,3}$, a graph with 4 vertices, one of degree 3 and 3 others of degree 1.

Two graphs are said to be *isomorphic* if they have equal number of components (vertices and edges) and also their edge connectivity is retained. In other words, an isomorphism of graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$ is a bijection $f : V_G \rightarrow V_H$ such that $uv \in E_G$ if and only if $f(u)f(v) \in E_H$.

1.1.2 Some graph classes

A graph class \mathcal{G} is the set of all graphs satisfying a certain property. In the following, we define all classes of graphs which will be seen throughout this thesis.

A graph $G = (V, E)$ is called *complete* and denoted by K_n where $|V| = n$, if for any pair $u, v \in V$, $uv \in E$. An undirected graph where any two vertices of it are connected by exactly one path is called *tree*, and a graph which is a collection of trees is called a *forest*.

A *bipartite graph* $G = (V, E)$ is an undirected graph in which the vertex set can be partitioned into two parts L and R such that the induced graph of each part makes an independent set. If in a bipartite graph, $N_G(u) = R$ for each vertex $u \in L$, it is called *complete bipartite graph* and is denoted by $K_{L,R}$. A *split graph* $G = (C \cup I, E)$ is an undirected graph where the vertex set $C \cup I$ is decomposable into a clique C and an independent set I .

A *star* is a tree where at most one vertex has a degree greater than 1 or, equivalently, it is isomorphic to $K_{1,\ell}$ for some $\ell \geq 0$. The vertices of degree 1 are called *leaves* of the star while the remaining vertex is called *center* of the star; for the case $\ell = 1$, we choose one vertex to be the center and the other to be the leaf (so there are not two leaves in $K_{1,1}$). A ℓ -star is a star of ℓ leaves; when $\ell = 0$, the star is called *trivial* and it is reduced to a single vertex (the center).

A *k-tree* is a graph which can be formed by starting from a k -clique and then repeatedly adding vertices in such a way that each added vertex has exactly k neighbors completely connected together (this neighborhood is a k -clique). A graph is a *partial k-tree* if it is a subgraph of a k -tree.

A graph is *planar* if it can be embedded in a plane. It means that it can be drawn on the plane in such a way that all the edge intersections are placed at the endpoints of edges.

An *interval graph* is a graph in which there exists a family of intervals on the real line and there is a bijection between the vertices of the graph and the family of intervals such that there is an edge in the graph if and only if the corresponding intervals have a non-empty intersection.

Circular-arc graphs are a natural generalization of interval graphs. A *circular-arc graph* is the intersection graph of a set of arcs on the circle. It has one vertex for each arc in the set, and an edge between every pair of vertices corresponding to arcs that intersect.

If for all cycles of four or more vertices of graph G , there is an edge that is not part of the cycle but connects two vertices of the cycle, the graph and the connected edge is called a *chordal graph* and a *chord* respectively. There are many characterizations of chordal graphs. One of them, known as Dirac's theorem, affirms that a graph G is chordal if and only if each minimal vertex separator of G is a clique. For any integer $k \geq 3$, a graph is called *k-chordal* if it has no induced cycle of length greater than k . Thus, chordal graphs are precisely the 3-chordal graphs. In particular the class of 4-chordal graphs contains another well known class of graphs called *weakly triangulated* graphs or also *weakly chordal*. This class is introduced in [102], in view of extending chordal graphs as the class with no chordless cycle on five or more vertices in $G = (V, E)$ or in its complement $\bar{G} = (V, \bar{E})$, or equivalently, the graph contains neither a *hole* nor an *anti-hole*.

Given a graph H , a graph is H -free, if it does not contain H as an induced subgraph. A *cograph* is a graph which can be formed by starting from a single vertex and by repeating application of complementation and vertex-disjoint union. These are precisely the P_4 -free graphs. A *line graph* of a graph G , denoted by $L(G)$ is a graph whose vertices represent the edges of G and two vertices of $L(G)$ are adjacent if and only if their corresponding edges share a common endpoint in G .

1.1.3 Tree decomposition

A *tree decomposition* of an original graph $G = (V, E)$, denoted by (T, B) where $T = (V_T, E_T)$ is a tree and B consists a subset $B_t \subseteq V$ associated with each node $t \in V_T$. The subsets B_t are called *bags* of the tree decomposition. (T, B) must satisfy the following conditions:

- 1– (vertex coverage) Every vertex of V belongs to at least one bag B_t .
- 2– (edge coverage) For every edge $e = uv \in E$, there is some bag B_t containing both u, v .
- 3– (coherence) For each $\{t_1, t_2, t_3\} \subseteq V_T$ where t_2 lies on the path from t_1 to t_3 in T , if a vertex $v \in (B_{t_1} \cap B_{t_3})$, then $v \in B_{t_2}$.

The *width* of a tree decomposition (T, B) , is termed by $width(T, B)$ and equals $\max_{t \in V_T} |B_t| - 1$. The *tree-width* of a graph G is the minimum width of any decomposition of G . A tree decomposition in which the underlying tree is a path, is called a *path decomposition* and the width parameter derived from a path decomposition is known as *path-width*.

Figure 1.1 presents two different tree decompositions for a given graph G . The tree in the middle part of the figure has width 3, while width of the other tree is 2.

A tree decomposition is converted easily into a *nice tree decomposition*, to decrease the possible transactions between bags [125]. A tree decomposition is nice if every node $t \in V_T$ be one of the following types:

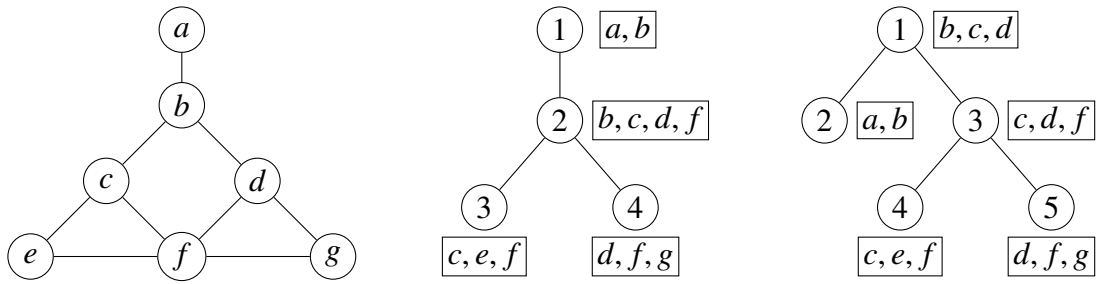


Figure 1.1: On the left, a graph $G = (V, E)$ is shown. Two tree decompositions of G are presented on the middle and right part. The nodes inside each rectangle show the associated bag.

- **Leaf:** a node with no children, $|B_t| = 1$.
- **Introduce:** a node with one child t' such that $B_t = B_{t'} \cup \{v\}$ for some $v \in V$.
- **Forget:** a node with one child t' such that $B_t = B_{t'} \setminus \{v\}$ for some $v \in V$.
- **Join:** a node with two children t' and t'' such that $B_t = B_{t'} = B_{t''}$.

Sometimes, instead of operating an algorithm on a given graph, we should operate it on a (not unique) tree decomposition of the original graph. For a given graph G and an integer k , it is NP-hard to determine whether G has a tree decomposition with width at most k [9]. However, if k is a fixed constant, the former decision problem is linear time solvable [27].

For more information about tree decomposition, we recommend [28, 125].

1.2 Computational complexity

An *instance* of a problem, includes an exact specification of the data which are involved in the problem. An *algorithm* for Π is a procedure which takes an instance as an input and gives an output. The *running time* of an algorithm for a specific input depends on the number of operations executed and the running time of an algorithm is the worst case input scenario, i.e. the greatest number of operations executed for some instance. Usually, instead of using the exact number of executed operations, we use the notation of "big- O ", "big- Ω ", "big- Θ ", "little- o " and "little- ω ", to express a bound for general running time. For two positive function f, g , the notations are defined as follows:

- $f(n) = O(g(n))$ means that there exist positive constants c and k , such that $f(n) \leq cg(n)$ for all $n \geq k$.
- $f(n) = \Omega(g(n))$ means that there exist positive constants c and k , such that $f(n) \geq cg(n)$ for all $n \geq k$.
- $f(n) = \Theta(g(n))$ means that $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

- $f(n) = o(g(n))$ if $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$ and $f(n) = \omega(g(n))$ if $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \infty$.

An algorithm is called *deterministic* if for each given input, its execution and output is unique, and an algorithm is *non-deterministic* if for the same input, it can exhibit different behaviors and executions.

1.2.1 NP-hardness and polynomial solvability

We study computational complexity theory to be able to categorize the problems into classes of running times that are achieved by some algorithms which solve them optimally.

Decision problems

A *decision problem* is a yes or no question on a finite set of inputs, i.e. the answer is yes or no for any instance of the set. Throughout this thesis, we use the following standard structure to define a new decision problem:

DECISION PROBLEM'S NAME

Input: An instance of the new decision problem.

Question: A yes-no question that relies on the input.

The class of all decision problems that can be solved by a non-deterministic algorithm in polynomial time is called NP and the class of decision problems that can be solved by a deterministic algorithm in polynomial time is called P. A decision problem D_1 polynomially *reduces* to a decision problem D_2 , if there exists an algorithm which takes an instance X_1 of D_1 as an input and produces an instance x_2 of D_2 in polynomial time such that, x_1 is a *yes*-instance of D_1 if and only if x_2 is a *yes*-instance of D_2 . The concept of NP-completeness was introduced in 1971 by Cook [55], when he proved that any problem in NP can be polynomially reduced to a famous decision problem SAT, which is defined as follows:

SAT

Input: A set \mathcal{C} of CNF clauses over a set \mathcal{X} of Boolean variables.

Question: Is there an assignment that satisfies all clauses of \mathcal{C} ?

Cook showed that all NP problems can be solved in polynomial time if SAT can be, and if any of the problems in this class is intractable, then SAT also can not be solved in polynomial time. John Hopcroft brought many scientists at the conference STOC in 1971 to discuss about whether an NP problem can be solved in polynomial time on a deterministic Turing machine. No one could answer this and the question of whether $P = NP$ is known as an open question now. In 1972, Karp proposed some reductions from some problems in NP to other problems in this class [118]. Thus, he introduced the term NP-complete for these problems which is defined as follows: A decision problem D is NP-complete if (i) $D \in NP$ and (ii) every problem D' in NP can be reduced in polynomial time to D . If the decision problem D just satisfies the condition

(ii), it is called NP-hard. After Karp's results, many well-known decision problems were proved NP-complete. Many of these problems are gathered by Garey and Johnson and presented in "Computers and Intractability" in 1979 [88].

Optimization problems

An *optimization problem* Π is a tuple $\langle I, F, opt, m \rangle$ where:

- I is the set of instances of the problem.
- $F(x)$ is a set of feasible solutions for instance $x \in I$.
- opt is the goal of the optimization problem (either maximization or minimization).
- m is a measure function that associates with any instance $x \in I$ and with any $y \in F(x)$ a natural number $m(x, y)$.

For an optimization problem, an *optimal solution* is a feasible solution that carries out the goal of the problem, it means that maximizes (or minimize) the measure function.

An optimization problem Π is called *hereditary*, if for any feasible solution S of a given instance x , each $S' \subseteq S$ is also a feasible solution. An optimization problem Π is called *anti-hereditary*, if for any feasible solution S of a given instance x , each $S' \supseteq S$ is also a feasible solution. An optimization problem is monotone if either it is hereditary or anti-hereditary.

Throughout this manuscript, we use the following standard structure to define a new optimization problem:

OPTIMIZATION PROBLEM'S NAME

Input: A description of an instance $x \in I$.

Solution: A description of a feasible solution $y \in F(x)$.

Output: Maximize or minimize (according to opt) the measure $m(x, y)$.

There is a relation between the optimization and decision variants of a problem. Given a maximization (resp. minimization) problem Π and by introducing a parameter k , we can define the corresponding decision problem D as follows: "Is there a solution of value at least (resp. at most) k ?". In the following, we propose some well-known optimization problems that are used many times throughout the manuscript.

MIN VERTEX COVER (MIN VC for short)

Input: A graph $G = (V, E)$.

Solution: A subset $S \subseteq V$ such that each $e \in E$ is incident to at least one $v \in S$.

Output: Minimize $|S|$.

MAX INDEPENDENT SET (MAX IS for short)

Input: A graph $G = (V, E)$.

Solution: A subset $I \subseteq V$ such that $G[I]$ contains no edges.

Output: Maximize $|I|$.

MIN DOMINATING SET (MIN DS for short)

Input: A graph $G = (V, E)$.

Solution: A subset $D \subseteq V$ such that $N_G[D] = V$.

Output: Minimize $|D|$.

MAX EDGE MATCHING (MAX EM for short)

Input: A graph $G = (V, E)$.

Solution: A subset $M \subseteq E$ such that none of the vertices of V is incident to more than one edge in M .

Output: Maximize $|M|$.

NP-Optimization (NPO) is the class of optimization problems where the associated decision problems are in NP. An optimization problem is polynomial-time solvable if there exists an algorithm which computes the optimal solution for every instance of the problem in polynomial time. The class of all optimization problems that are polynomial-time solvable is called P-Optimization (PO).

For most of our NP-hardness proofs in the next chapters, we use the structure of a bipartite graph which can be made from an instance $I = (\mathcal{C}, \mathcal{X})$ of SAT, using the following definition:

Definition 1.1. Let $I = (\mathcal{C}, \mathcal{X})$ be an instance of SAT with clause set $\mathcal{C} = \{c_1, \dots, c_m\}$ and variable set $\mathcal{X} = \{x_1, \dots, x_n\}$, the corresponding variable-clause-graph $BP = (X \cup C, E(BP))$ is a bipartite graph with $C = \{c_1, \dots, c_m\}$, $X = \{x_1, \dots, x_n\}$ and $E(BP) = \{c_j x_i : x_i \text{ or } \neg x_i \text{ is literal of } c_j\}$.

Let us here introduce two variants of SAT which will be used for our NP-hardness proofs later.

- The first is known as 2-BALANCED 3-SAT, denoted by $(3, B2)$ -SAT. An instance $I = (\mathcal{C}, \mathcal{X})$ of $(3, B2)$ -SAT is given by a set \mathcal{C} of CNF clauses defined over a set \mathcal{X} of Boolean variables such that each clause has exactly 3 literals, and each variable appears exactly 4 times in \mathcal{C} , twice negated and twice unnegated.
- The other problem is 4-BOUNDED PLANAR 3-CONNECTED SAT (4P3C3SAT for short). Let $I = (\mathcal{C}, \mathcal{X})$ be an instance of 4P3C3SAT, where $\mathcal{C} = \{c_1, \dots, c_m\}$ and $\mathcal{X} = \{x_1, \dots, x_n\}$ are the clause set and variable set of I , respectively, then each clause has exactly 3 literals and each variable occurs in at most 4 clauses (at least one time negated and one time unnegated). Moreover, the corresponding variable-clause-graph $BP = (X \cup C, E(BP))$ with $C = \{c_1, \dots, c_m\}$, $X = \{x_1, \dots, x_n\}$ and $E(BP) = \{c_j x_i : x_i \text{ or } \neg x_i \text{ is a literal of } c_j\}$, is planar bipartite of maximum degree 4.

In [21, 127] it is proved that, deciding whether an instance of $(3, B2)$ -SAT and 4P3C3SAT respectively, is satisfiable is NP-complete.

By Definition 1.1, the associated *variable-clause-graph* $BP = (C \cup X, E(BP))$ corresponding to an instance I of $(3, B2)$ -SAT (resp. 4P3C3SAT) is bipartite, the left part C is 3-regular and the right part X is 4-regular (resp. is planar bipartite of maximum degree 4.) In this chapter, we always assume that the planar graph comes with an embedding in the plane. Informally, we are building a new graph (corresponding to each problem) by putting some variable (resp. clause) gadgets instead of vertices x_i (resp. c_j) of BP which satisfy the following two conditions:

- the construction distinguishes between the cases that a variable x_i appears positively and negatively in some clauses,
- the construction preserves planarity.

Using a Karp reduction from an NP-complete problem, we are able to prove the NP-hardness for a new problem. However, some other reductions are "*approximation preserving reductions*" which help us to prove there is no polynomial algorithm obtaining a certain ratio for a given problem which will be explained in the next section.

1.2.2 Approximation of NP optimization problems

Unfortunately, most interesting optimization problems are NP-hard. Thus, under the widely believed conjecture that $P \neq NP$, computing their exact solution is excessively time consuming. So, instead of spending exponential time to find the optimal solution, the goal is to give an algorithm that runs in polynomial time and outputs a solution whose value is closed to the optimum value with a certain ratio called the approximation ratio, which gives a guarantee for the quality of the given solution.

A ρ -approximation algorithm for an optimization problem is a polynomial time algorithm that for every instance of the problem, gives a solution whose value will not be more (or less, depends on whether opt is maximize or minimize) than a factor ρ times the value the optimal solution. More formally, for a given NP-optimization problem Π , \mathcal{A} is an approximation algorithm for Π , if \mathcal{A} returns a feasible solution $\mathcal{A}(I)$ for each instance I . Let $m(I, \mathcal{A}(I))$ and $opt(I)$ be the value of the approximate solution $\mathcal{A}(I)$ and the value of the optimal solution respectively, then the *approximation ratio*² of \mathcal{A} with respect to I for optimization problem Π is: $r_{\mathcal{A}}(I) = \frac{m(I, \mathcal{A}(I))}{opt(I)}$. For a constant $\rho \geq 1$ in the case of a minimization problem (resp. $\rho \leq 1$ in the case of a maximization problem), \mathcal{A} is an ρ -approximation algorithm, if $\rho \geq r_{\mathcal{A}}(I)$ (resp. $\rho \leq r_{\mathcal{A}}(I)$) for every instance I of minimization (resp. maximization) problem Π . We say also that Π is ρ -approximable, if \mathcal{A} is an approximation algorithm with ρ ratio.

In addition, regarding to the theoretical issues for approximation, we need to answer what is the best performance ratio of any polynomial-time approximation algorithm for a special NPO

²Performance ratio is defined by $\frac{|m(I, \mathcal{A}(I)) - opt(I)|}{opt(I)}$.

problem. Motivated by this theoretical question, a variety of classes for NPO problems were defined, depending on their approximability properties.

An *approximation scheme* for a minimization (resp. maximization) problem Π , is a family of polynomial-time approximation algorithms with ratio $(1 + \epsilon)$ (resp. $(1 - \epsilon)$) for any $\epsilon > 0$ given in the input. A *polynomial time approximation scheme* PTAS for problem Π is an approximation scheme with a polynomial time complexity with respect to the input size. A *fully polynomial time approximation scheme* FPTAS for problem Π is an approximation scheme with a polynomial time complexity with respect to the input size and also with respect to $\frac{1}{\epsilon}$. Hence, for a PTAS it would be acceptable to have an algorithm with time complexity $n^{\frac{1}{\epsilon}}$ where n is the input size, but it is not admissible for an FPTAS, since it is polynomial in the size of the input but it is exponential in $\frac{1}{\epsilon}$. An *efficient polynomial time approximation scheme* EPTAS for problem Π is an approximation scheme such that for a given instance of length n , one can find a solution in time $f(\epsilon)p(n)$, where p is a polynomial and f is arbitrary. So, every EPTAS is also a PTAS and every FPTAS is also an EPTAS.

Given an NPO problem Π , it belongs to the class APX if there is a polynomial-time algorithm with approximation ratio bounded by a constant. An NPO problem Π belongs to the class *log-APX* (resp. *poly-APX*) if there is a polynomial-time algorithm with approximation ratio bounded by $c \cdot \log(|x|)$ (resp. $c \cdot p(|x|)$ where p is a polynomial function), for some constant c and where $|x|$ is the size of the instance.

Some examples for each of the above classes are proposed in [13]. We can deduce the following inclusions:

$$PO \subseteq FPTAS \subseteq EPTAS \subseteq PTAS \subseteq APX \subseteq \text{log-APX} \subseteq \text{poly-APX} \subseteq NPO$$

It is possible that the NPO problems that are reducible to each other in polynomial time, have different approximability properties. For instance, the two optimization problems MAX IS and MIN VC, are equivalent to each other in complexity point of view (both are NP-complete and $S \subseteq V$ is an independent set of $G = (V, E)$ if and only if $\bar{S} = V \setminus S$ is a vertex cover of G) but they have quite different properties when it comes to approximation: MAX IS is non-approximable within $n^{\epsilon-1}$ unless $P = NP$, where n is the number of vertices of the input graph and $0 < \epsilon < 1$ (i.e. there is no polynomial-time algorithm which guarantees an approximation ratio better than $O(n)$ for all the instances of the problem) [164], while one can find a $\frac{1}{2}$ -approximation ratio for MIN VC by repeatedly taking both endpoints of an edge into the solution and deleting adjacent edges.

An NPO problem Π is PTAS-hard (resp. APX-hard), if it is not in FPTAS (resp. PTAS). In order to construct reductions between NPO problems which preserve approximability, many researches have been done and different types of approximation-preserving reductions have been defined over the years, such as AP-reduction, A-reduction, F-reduction, P-reduction, R-reduction, L-reduction, etc [114, 12]. In the following, we propose a more practical approximation-preserving reduction which is called L-reduction, and proposed by Papadimitriou and Yannakakis [147]:

Definition 1.2. Let Π and Π' be two optimization problems in NPO. There is an L-reduction with parameters α and β with $\alpha, \beta > 0$ from Π to Π' if:

- (i) For each instance I of Π , we can compute in polynomial-time an instance I' of Π' such that $opt_{\Pi'}(I') \leq \alpha opt_{\Pi}(I)$.
- (ii) Given a solution S' of I' of value $m(I', S')$, we can compute a solution S of I of value $m(I, S)$ in polynomial time such that $|opt_{\Pi}(I) - m(I, S)| \leq \beta |opt_{\Pi'}(I') - m(I', S')|$.

According to Definition 1.2, if Π and Π' are two maximization problems and if Π' is ρ -approximable, there is a $(\rho\beta(\alpha - 1) + 1)$ -approximation algorithm for Π . Moreover, if there is an L-reduction from Π to Π' , and if Π' is in PTAS (resp. belongs to APX), then Π is in PTAS (resp. belongs to APX). Equivalently, if Π is PTAS-hard (resp. APX-hard), then Π' is also PTAS-hard (resp. APX-hard).

For more information about approximation algorithms, we recommend [106, 158].

1.2.3 Parameterized complexity

When we have to handle an NP-hard problem, it can be interesting to investigate parameterized complexity. Usually, we always express the running time of an algorithm depending on the size of the instance (for example, in graphs, it is often the number of vertices or edges). Parameterized complexity is a new approach for handling NP-hard problems. Within the last 20 years, a viewpoint was introduced by Downey and Fellows [78], where one can measure the time complexity of an algorithm not only in terms of the input size but also respect to multiple parameters of the input or output. Formally, a parameterized problem is a language $L \subseteq \Sigma^* \times \mathbb{N}$, where Σ is a finite set of alphabet and the second component is the parameter of the problem.

A lot of problems admit an algorithm with running time bounded by $c|x|^k$ where x is an input, c is a constant and k is the size of solution. As an example, let us consider the decision problem CLIQUE which is defined as follows:

CLIQUE

Input: A graph $G = (V, E)$ and an integer k .

Question: Does G contain a clique (complete subgraph) of size at least k ?

Deciding whether there is a clique of size at least k in a graph $G = (V, E)$ can be easily solved in $O(|V|^k)$ by checking any subset of size k in G . More generally, the set of all parameterized problems which admit some algorithms with running time $O(f(k)|x|^k)$, is called XP, where f is an arbitrary computable function and $|x|$ is the size of the input. Unlike that, there are some parameterized problems for which it is unlikely that algorithms with such a running times exist. This class of problems is called XP-hard (or not in XP). As an example of these problems, let us consider a decision problem in coloring, namely to if a given graph G has chromatic number at most k . The problem is formally defined as follows:

k -COLORING**Input:** A graph $G = (V, E)$ and a fixed integer k .**Question:** Is G k -colorable? (i.e. is there a proper coloring with at most k colors of G ?)

It is proved that the k -COLORING problem in general graphs for fixed $k \geq 3$ is NP-hard assuming $P \neq NP$ [134]. Now, suppose k -COLORING is in XP— this means that for a given graph G of order n and a fixed integer $k \geq 3$, there is an algorithm with running time $O(f(k)n^k)$, which contradicts the hardness of k -COLORING. However, it can be even more interesting to design an algorithm with a running time that separates the size of the instance x by a parameter k . In that way, the class of all parameterized problems which can be decided by an algorithm with running time $f(k)|x|^{O(1)}$ time is called FPT, where $|x|$ is the input size and f is a computable function and k is a parameter. Such an algorithm is called *fixed parameter tractable algorithm* (fpt-algorithm for short). Hence, if k is fixed and small, such problems can be considered "tractable" whereas, they were considered "intractable" in the traditional classification.

As well as the problems that belong to FPT, there are some problems that belong to XP, but unbelievably admit an fpt-algorithm. In this regard, we need a notion of reduction which transfers a negative evidence for fixed-parameter tractability from one parameterized problem to another one.

Definition 1.3 ([66, 67, 68]). *Let Π, Π' be two parameterized problems. A parameterized reduction from Π to Π' is an algorithm that, for a given instance (x, k) of Π , gives an instance (x', k') of Π' such that:*

- (x, k) is a yes-instance of Π if and only if (x', k') is a yes-instance of Π' ,
- $k' \leq g(k)$ for some computable function g ,
- the running time is $f(k), |x|^{O(1)}$ for some computable function f .

So, if there is a parameterized reduction from Π to Π' and Π' is FPT, then Π is FPT as well, and equivalently, if Π is not FPT, then Π' is not FPT either.

Most of the NP-hard problems (all that belong to NP) are equivalent to each other with respect to classical polynomial-time reductions, but this does not seem to be true for parameterized point of view. Downey and Fellows introduced a hierarchy to classify parameterized problems according to their hardness [70, 68, 66, 67, 69]. They mentioned that NP-hard problems may appear in different classes of parameterized intractability. For example two NP-complete problems CLIQUE and INDEPENDENT SET with respect to the natural parameter (the size of the optimal solution) are W[1]-hard [36, 140], while the other NP-complete problem DOMINATING SET is W[2]-hard with respect to the same parameter [70]. It shows that, unlike in the case of classical complexity, INDEPENDENT SET and DOMINATING SET occupy different levels of this hierarchy. In the following we introduce the W-hierarchy proposed by Downey and Fellows:

$$P \subseteq \text{FPT} \subseteq \text{W}[1] \subseteq \text{W}[2] \cdots \subseteq \text{W}[t] \cdots \subseteq \text{XP}.$$

A parameterized problem is $W[t]$ -hard if every problem of $W[t]$ FPT-reduces to it. It is $W[t]$ -complete if it is $W[t]$ -hard and belongs to $W[t]$. In the following, we want to show how we can prove that a parameterized problem belongs to a specific class in the parameterized hierarchy.

In order to prove that a problem Π belongs to a $W[t]$ -class, we always pick a problem Π' already in the class and try to construct a reduction from Π to Π' . Usually finding this kind of reduction is difficult since the two problems can be very different. In the following we introduce two types of parameterized problems, namely *circuit satisfiability* and *Turing machine*, which help us to construct such parameterized reductions.

We need first to define preliminary concepts. A Boolean circuit is a directed acyclic graph where the nodes are labeled in the following way:

- every node of in-degree 0 is an input node,
- every node of in-degree 1 is a negation node,
- every node of in-degree ≥ 2 is either an and-node or an or-node.

Additionally, there is exactly one node of out-degree 0 called output. A node with in-degree bounded by a constant is said to be small, and otherwise it is called large. The *weft* of a boolean circuit is the maximum number of large nodes on a path from an input to the output. The depth is the maximum number of all nodes on a path from an input to the output. Assigning true-false values to the input nodes determines the value of every node in the obvious way. Particularly, if a given assignment to the input nodes leads to a true value for the output node, then we say that the assignment *satisfies* the circuit and the weight of this assignment is the number of input nodes that are assigned true.

Deciding if a boolean circuit has a satisfying assignment is an NP-complete problem since (i) SAT is a special case of the problem and (ii) it can be easily checked in polynomial time if a given assignment satisfies the circuit. A parameterized version of finding a satisfying assignment denoted by $W_{\text{EFT-}t}$ CIRCUIT SATISFIABILITY (W - t CS for short) defined in the following way.

$W_{\text{EFT-}t}$ CIRCUIT SATISFIABILITY (W - t CS)

Input: A boolean circuit C with constant depth and weft at most t and an integer k .

Question: Is there a truth assignment of weight k that satisfies C ?

Definition 1.4. For $t \geq 1$, a parameterized problem Q belongs to $W[t]$ if there is a parameterized reduction from Q to W - t CS.

A way to prove that a parameterized problem belongs to $W[t]$ is to construct a parameterized reduction from the problem to a problem known to be in $W[t]$. Here, we propose two parameterized reductions from the decision problems INDEPENDENT SET and DOMINATING SET to W -1CS and W -2CS respectively, which are established in [58].

INDEPENDENT SET belongs to W[1]:

Let $G = (V, E)$ with $V = \{v_1, \dots, v_n\}$ be an instance of INDEPENDENT SET, we construct a boolean circuit $C = (V_C, A_C)$ as an instance of W-1CS as follows:

- Introduce n input nodes a_1, \dots, a_n of in-degree 0 corresponding to vertices v_1, \dots, v_n .
- For each node a_i , introduce a node b_i labeled by "not" and add an arc (a_i, b_i) to A_C .
- For any two vertices v_i, v_j in V that $v_i v_j \in E$, create a node c_{ij} labeled by "or" and add two arcs (b_i, c_{ij}) and (b_j, c_{ij}) to A_C .
- Add a node d labeled by "and" and add the arcs (c_{ij}, d) for all c_{ij} to A_C .

Using a simple example in Figure 1.2, we illustrate the construction. Notice that, the depth of the boolean circuit C is 4, and the boolean circuit has weft 1 since, the only large node in C is node d . Moreover, notice that there is an independent set of size at least k in G if and only if there is a truth assignment of weight k that satisfies C .

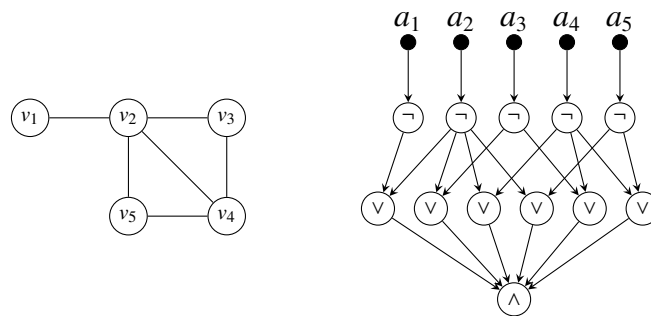


Figure 1.2: The construction of the boolean circuit $C = (V_C, A_C)$ from an instance $G = (V, E)$ of INDEPENDENT SET.

DOMINATING SET belongs to W[2]:

Let $G = (V, E)$ with $V = \{v_1, \dots, v_n\}$ be an instance of DOMINATING SET, we construct a boolean circuit $C = (V_C, A_C)$ as an instance of W-2CS as follows:

- Introduce n input nodes a_1, \dots, a_n of in-degree 0 corresponding to vertices v_1, \dots, v_n .
- For each node a_i , introduce a node b_i labeled by "or" and for each $v_j \in N_G[v_i]$, add the arc (a_i, b_j) to A_C .
- Add a node c labeled by "and" and add the arcs (b_i, c) for all $1 \leq i \leq n$ to A_C .

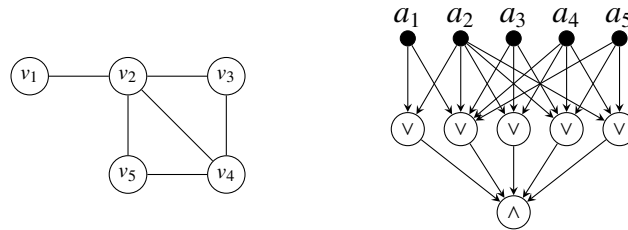


Figure 1.3: The construction of the boolean circuit $C = (V_C, A_C)$ from an instance $G = (V, E)$ of DOMINATING SET.

Using a simple example in Figure 1.3, we illustrate the construction. Notice that, the depth of the boolean circuit C is 2, and the boolean circuit has weft 2, since all b_i and c have in-degree greater than 2. Moreover, notice that there is a dominating set of size at most k in G if and only if there is a truth assignment of weight k that satisfies C .

In the above examples, we made two different parameterized reductions from INDEPENDENT SET and DOMINATING SET to W-1CS and W-2CS which are known to belong to $W[1]$ and $W[2]$ respectively. In the following, we introduce a class of different *Turing machine* problems which belong to levels of the parameterized complexity hierarchy.

A *Turing machine* is a mathematical model of computation that defines an abstract machine, which was invented in 1936 by Alan Turing. A Turing machine consists of a tape of infinite length where read and write operations can be performed. The tape consists of infinite cells on which each cell either contains input symbol or a special symbol called blank. It also consists of a head pointer which points to cell currently being read and it can move in both directions. A Turing machine is expressed as a 7-tuple $(Q, T, B, \Sigma, \delta, q_0, F)$ where:

- Q is a finite set of states
- T is the tape alphabet (symbols that can be written on the tape)
- B is blank symbol (every cell is filled with B except where the input symbols are written onto the tape initially)
- Σ is the tape alphabet
- δ is a transition function which maps $Q \times T \rightarrow Q \times T \times \{L, R\}$. Depending on its present state and present tape alphabet (pointed by head pointer), it will move to a new state, change the tape symbol (this is optional) and move the head pointer to either left or right. Multi-tape Turing Machines have multiple tapes where each tape is accessed with a separate head. Each head can move independently of the other heads. The transition function for a k -tape Turing Machine is modeled by $Q \times T^k \rightarrow Q \times T^k \times \{L, R\}^k$.
- q_0 is the initial state
- F is the set of final states. If any state of F is reached, input string is accepted.

A Turing machine is *deterministic* if the set of rules prescribes at most one action to be performed for any given situation. By contrast, in a *nondeterministic* Turing machine, the set of rules may prescribe more than one action to be performed for any given situation. In the following we introduce two types of nondeterministic Turing machine problems which are known in parameterized complexity.

SHORT NONDETERMINISTIC TURING MACHINE

Input: A single-tape nondeterministic Turing machine M , a word x over the alphabet of M and a positive integer k .

Question: Is there a computation of M on x that reaches a final accepting state in at most k steps?

In [37, 72], it is proved that SHORT NONDETERMINISTIC TURING MACHINE problem is $W[1]$ -complete. In order to show that a parameterized problem Π belongs to $W[1]$, one can devise a parameterized reduction from Π to the SHORT NONDETERMINISTIC TURING MACHINE problem (see some examples in [42]). Turing machines can be also useful in constructing membership results for the class $W[2]$. In [43] it is proved that a natural variation of the short nondeterministic Turing machine problem in which the machine has many read/write tapes is $W[2]$ -hard. This version of the problem is defined formally as follows:

SHORT MULTI-TAPE NONDETERMINISTIC TURING MACHINE

Input: A multi-tape nondeterministic Turing machine M , a word x over the alphabet of M and a positive integer k .

Question: Is there a computation of M on x that reaches a final accepting state in at most k steps?

In order to show that a parameterized problem Π belongs to $W[2]$, one can devise a parameterized reduction from Π to SHORT MULTI-TAPE NONDETERMINISTIC TURING MACHINE problem (see some examples in [42]). Finally, a special kind of Turing machine problems is named BOUNDED NONDETERMINISTIC TURING MACHINE proposed in [42] to be useful in establishing membership results for the class $W[t]$.

There are a lot of fixed-parameter tractable problems, which have FPT algorithms with running times like $O^*(2^k)$, $O^*(3^{k \log k})$, $O^*(2^{k^2})$, \dots ³. Hence, we need an inner hierarchy for the FPT problems to give asymptotic lower bounds for running times of these problems. The assumption that $FPT \neq W[1]$ seems too weak to achieve this goal. Therefore two stronger (widely believed) assumption were introduced.

Let us here recall the Satisfiability problem. An instance of SAT consists of a CNF-formula over the variables x_1, \dots, x_n , i.e. a conjunction (AND) of clauses, where each clause is a disjunction (OR) of some literals, where a literal is a variable x_i or the negation of the variable. The output is yes or no according to whether there exists a truth assignment for the formula i.e. an assignment

³Throughout this manuscript, the O^* notation is extensively used in the running times of parameterized algorithms that suppresses polynomial factors in the input size. For example, $O^*(3^k)$ instead of $O(\text{poly}(n)3^k)$.

of true/false to each variable such that the formula being true. By restricting the number of variables appearing in each clause to some constant q , we arrive at the q -SAT problem. The q -SAT problem is NP-complete for $q \geq 3$. So, we do not expect to find a polynomial algorithm for the problem. Moreover, by trying all possible true/false assignments, we can solve any q -SAT problem in $O^*(2^n)$ time, where n is the number of variables. Additionally, no one has proposed an algorithm for 3-SAT that is substantially faster than the brute-force algorithm.

The *Exponential Time Hypothesis* (ETH for short) is a conjecture stating that there is no sub-exponential algorithm in the number of variables for 3-SAT. It means that an n -variable 3-SAT cannot be solved in time $2^{o(n)}$. Generally, ETH allows us to show some significant results in different forms. For example, one can prove that a problem cannot be solved in time $2^{o(n)}$, or a parameterized problem cannot be solved in time $f(k)n^{o(k)}$, or an FPT problem does not admit a $2^{o(k)}n^{O(1)}$ -time algorithm by (assuming ETH). The *Strong Exponential Time Hypothesis* (SETH for short) is a conjecture stating that general SAT can not be solved in running time $O((2 - \epsilon)^n)$ for any constant $\epsilon > 0$. In other words, let δ_q be the infimum of the set of constants c for which there exists an algorithm solving q -SAT in time $O^*(2^{cn})$ for $q \geq 3$. From ETH, we deduce that $\delta_3 > 0$, and from SETH we derive $\lim_{q \rightarrow \infty} \delta_q = 1$.

These two conjectures were first introduced by Impagliazzo and Paturi [110]. In order to propose some negative results, we can use ETH and SETH, via a particular hardness reduction, to show that there is not any algorithm of some specific running time. Some examples of using one of these conjectures in order to prove a lower bound for the time complexity of some problems are presented in [132, 133]. The reader can find more information about parameterized complexity in [71, 145, 58].

Extension of Minimal and Maximal Graph Problems

Contents

| | | |
|------------|--|-----------|
| 2.1 | Extension optimization problems | 49 |
| 2.2 | Decision problems associated with extension optimization problems | 50 |
| 2.3 | Price of extension | 53 |
| 2.4 | Max-min and min-max optimization problems in graphs | 54 |
| 2.5 | Overview | 57 |

This manuscript presents a general study of the behavior of graph optimization problems when a partial solution is given, generally known as *extension problems*. Several extension problems have already been studied in the literature— such problems generally involve optimizing an objective function among the feasible solutions containing a partial solution and/or excluding some other elements. For extension problems containing a partial solution (also called completion problems), existing results dealt with coloring [23, 109], scheduling problem [84], routing problems [7, 8, 113] and many other combinatorial problems completion have been studied in [64, 161]. A series of researches have been done related to the well-known PRECOLORING EXTENSION problem on different graph classes, in which, a graph and a partial proper coloring are given in advance, and it asks to find a proper coloring of the whole graph extending the partial one and using the same colors [139, 109, 146, 14, 159, 138, 23]. For example, in [139, 109], it is shown that the problem on proper interval graphs is NP-hard when the number of colors of the precolored set is unbounded. It is proved that PRECOLORING EXTENSION problem on general interval graphs is solvable in polynomial time when each color appears only once in the precolored set, while it is NP-hard if each color of the precolored set used twice [23]. For bipartite graphs, the NP-completeness result holds if the precolored set comprises 3 vertices of distinct colors.

An extension variant of the SHORTEST CYCLE problem consists of a weighted graph $G = (V, E, w)$ together with a set of specified elements $U \subseteq (E \cup V)$ and the goal is to extend U to a simple cycle with minimum weight. The problem is NP-hard but is in FPT by the natural parameter (the number of vertices or edges in the shortest cycle) [24]. A generalized version of

the TRAVELLING SALESMAN PROBLEM (TSP for short) in the framework of extension problems is known as the GENERAL ROUTING PROBLEM (GRP for short), and has been considered from an approximability point of view. The goal of the GRP for a given graph $G = (V, E)$ and two subsets $V' \subseteq V$ and $E' \subseteq E$, is to find a cycle of minimum cost in $G = (V, E)$ which visits all the vertices in a required subset V' exactly once and covers all the edges in a required subset E' at least once. In [113], using Christofides' algorithm for the TSP with triangle inequality, the authors presented a $\frac{3}{2}$ approximation algorithm for GRP. Further, the ORDERED CLUSTER TRAVELING SALESMAN PROBLEM (OCTSP for short) have been studied in the literature [7, 8, 91, 99]. In this problem, the input contains a weighted graph $G = (V, E, w)$ plus an ordered partition of vertices of G into k clusters C_1, \dots, C_k . The goal of OCTSP is to find a simple tour of all vertices with minimum weight such that it first visits all vertices in C_1 , then all vertices in C_2, \dots , and finally all the vertices in cluster C_k . Other types of extension involve optimization problems subject to a forcing graph or conflict graph [62], where for an instance dealing with the shortest path problem, a constraint between distinct pair $\{u, v\}$ of vertices is called a conflict (resp. forcing) constraint if at most (resp. at least) one vertex among u, v is traversed by the path.

In this thesis, a large number of hereditary and anti-hereditary graph optimization problems will be studied under a framework of extension problems. Formally, an optimization problem $\Pi = \langle I, F, opt, m \rangle$ is **hereditary** (resp. **anti-hereditary**) if for any instance $x \in I$, the set of feasible solutions $F(x)$ is closed under *set-inclusion* (resp. *set-exclusion*), i.e. for any instance $x \in I$ and any feasible solution $y \in F(x)$, $y' \in F(x)$ for every $y' \subseteq y$ (resp. $y' \supseteq y$). In our setting the instance x is a graph $G = (V, E)$ and $F(G) \subseteq 2^V$ or $F(G) \subseteq 2^E$ depending on the context. For example, MAX IS is a graph hereditary problem, since for any given graph $G = (V, E)$, $S' \subseteq S$ is an independent set if $S \subseteq V$ is an independent set, and MIN VC is a graph anti-hereditary problem, since for any given graph $G = (V, E)$, $S' \supseteq S$ is a vertex cover of G if $S \subseteq V$ is too.

In our framework, we focus on inclusion-wise minimal and maximal solutions. Given a graph optimization problem $\Pi = \langle I, F, opt, m \rangle$, for an instance $G \in I$ and a set of feasible solutions $F(G)$, the set of inclusion-wise minimal (resp. maximal) feasible solution is denoted by $\mu(G)$ (resp. $\psi(G)$) and is defined as follows:

- $\mu(G) = \{y \in F(G) : ((y' \subseteq y) \wedge (y' \in F(G))) \rightarrow y' = y\}$
- $\psi(G) = \{y \in F(G) : ((y' \supseteq y) \wedge (y' \in F(G))) \rightarrow y' = y\}$

Let Π be an anti-hereditary (resp. a hereditary) optimization problem. For an instance $G \in I$ of Π , an instance of the extension problem Π' includes G together with a pre-solution U , that is a subset of an appropriately chosen part of the input for each particular problem. For instance, for the MIN VC problem, the instance contains a graph $G = (V, E)$ and a subset $U \subseteq V$, while for MAX EM, the instance includes a graph $G = (V, E)$ and the pre-solution $U \subseteq E$. In this framework, the pre-solution U when Π is an anti-hereditary (resp. a hereditary) problem, is called a *forced set* (resp. *co-forbidden set*).

In this chapter, a formal definition of any kind of optimization / decision problems within the extension framework is presented. We also try to review all the related studies in each section.

We finish this chapter with an brief overview of the results and problems which will be presented in the following chapters.

2.1 Extension optimization problems

Here, we define a graph optimization problem related to the extension problems. Let Π be an anti-hereditary (resp. hereditary) graph optimization problem. The extension optimization problem Π' denoted by $\text{MIN EXT } \Pi$ (resp. $\text{MAX EXT } \Pi$) consists of an instance G of Π together with a **forced set** (resp. a **co-forbidden set**¹) $U \subseteq V$ (or E depending the context) as an input, and the goal of Π' is to find a minimal (resp. maximal) solution S such that $U \subseteq S$ (resp. $U \supseteq S$). For example, for two problems MIN VC and MAX IS , the extension variants are formally defined as follows:

MIN EXT VC

Input: A graph $G = (V, E)$ together with a forced set $U \subseteq V$.

Solution: A minimal vertex cover $S \subseteq V$ such that $U \subseteq S$.

Output: Minimize $|S|$.

MAX EXT IS

Input: A graph $G = (V, E)$ together with a co-forbidden set $U \subseteq V$.

Solution: A maximal independent set $S \subseteq V$ such that $S \subseteq U$.

Output: Maximize $|S|$.

Here, it is important to notice that minimality and maximality are considered with respect to the whole set of feasible solutions. Hence, for an instance (G, U) of MIN EXT VC , a subset $S \supseteq U$ is a minimal vertex cover if $S - v$ is not a vertex cover for any $v \in S$. Another extension problem associated to vertex cover is defined by $S \supseteq U$, where S is a vertex cover and $S - v$ is not a vertex cover for any $v \in S \setminus U$. In this latter case, we restrict all the feasible solutions to contain U , and the goal is to optimize the solution part out of U ; this variant has been studied in [161, 64]. Weller et al. defined a *residue variant* of the extension problems, where for a feasible solution S of the problem, the goal is to optimize the residue part $(S \setminus U)$ [161]. They studied some classical optimization problems together with the description of *residue-approximation* in the framework of extension problem. In [64], authors added another constraint to the problem, in which, for an instance (G, U, F) of a graph problem Π , any feasible solution S contains all members of U whilst $S \cap F = \emptyset$. Many graph optimization problems have been studied in this framework from computational complexity and (in-)approximability points of view. Regarding another type of extension problem considered in [57], for a graph problem Π in this framework, the instance consists of a graph $G = (V, E)$ together with a partition Π_V (resp. Π_E) of its vertices (resp. of its edges), and any solution S containing an element (vertex or edge) of any partition must also contain all the other ones. Let us now give an example illustrating our framework.

¹ \bar{U} is called forbidden set.

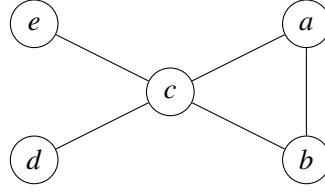


Figure 2.1: The graph $G = (V, E)$ as a part of the instances of MAX EXT IS and MIN EXT VC .

Example 2.1. Consider the graph given in Figure 2.1 as an example for MIN EXT VC and MAX EXT IS . First, suppose G together with the forced set $U = \{a, b\}$ is an instance of MIN EXT VC . The only minimal vertex cover containing $\{a, b\}$, is $\{a, b, e, d\}$, so the size of optimal solution of MIN EXT VC for the instance (G, U) is 4. However, for the extension framework proposed by Weller et al. [161] (focusing just on the residue part), the optimal vertex cover containing U is $\{a, b, c\}$. Now, suppose the graph G together with the co-forbidden set $U = \{c, d, e, a\}$ is an instance of MAX EXT IS . In this case, there are 2 maximal independent sets included in U namely, $\{a, e, d\}$, and $\{c\}$. Thus, the size of optimal solution of MAX EXT IS for the instance (G, U) is 3.

We now define formally $\text{ext}_\Pi(G, U)$ for hereditary and anti-hereditary problems. Let Π be an anti-hereditary graph problem, let $G = (V, E)$ be a simple graph and let $U \subseteq V$ or E (depending on the context). Then $\text{ext}_\Pi(G, U)$ is a subset $S \subseteq V$ or E (depending on the context) of minimal solutions of Π such that $S \supseteq U$. Now let Π be a hereditary graph problem, let $G = (V, E)$ be a simple graph and let $U \subseteq V$ or E (depending on the context). Then $\text{ext}_\Pi(G, U)$ is a subset $S \subseteq V$ or E (depending on the context) of maximal solutions of Π such that $S \subseteq U$.

Sometimes, for an instance (G, U) of a graph extension problem Π , $\text{ext}_\Pi(G, U) = \emptyset$ (i.e. there is no feasible solution), which makes the question of the existence of such extensions interesting. However, if one starts with a trivial pre-solution, then the question is usually easy. In the next section, we define decision problems associated with our framework.

2.2 Decision problems associated with extension optimization problems

The question as to whether a given partial solution to a problem can be extended reasonably has been developed in many algorithmic approaches for optimization problems. For instance, when enumerating minimal dominating sets of a graph $G = (V, E)$, one usually arrives at the problem to decide for a vertex set $U \subseteq V$, if there exists a *minimal* dominating set S with $U \subseteq S$. For a given hereditary (resp. anti-hereditary) graph problem Π , the *extension decision problem* for Π is written $\text{EXT } \Pi$. Informally, an input of $\text{EXT } \Pi$ consists of an instance $G = (V, E)$ of Π , together with a pre-solution $U \subseteq (V \text{ or } E \text{ depending the context})$, and the associated decision problem asks if there is a minimal (resp. maximal) solution S for an anti-hereditary (resp. hereditary) problem Π such that $U \subseteq S$ (resp. $U \supseteq S$). For example, for two problems vertex cover and independent set, EXT VC and EXT IS formally defined as follows:

EXT VC

Input: A graph $G = (V, E)$ together with a forced set $U \subseteq V$.

Question: Is there a minimal vertex cover S such that $U \subseteq S$?

EXT IS

Input: A graph $G = (V, E)$ together with a co-forbidden set $U \subseteq V$.

Question: Is there a maximal independent set S such that $U \supseteq S$?

In Remark 4.1, we will show that (G, U) is a yes-instance of EXT VC if and only if $(G, V \setminus U)$ is a yes-instance of EXT IS. To be clear, let us give an example.

Example 2.2. Consider the graph G illustrated in Figure 2.2, and suppose that $(G, \{g, h, f\})$ and $(G, \{g, a, e\})$ are two instances of EXT VC. It is clear that $\{g, h, f\}$ can be extended to a minimal vertex cover in G , as $\{g, h, f, d, b\}$ is a minimal vertex cover containing U , but $(G, \{g, a, e\})$ is a *no*-instance of EXT VC, i.e. there is no minimal vertex cover of G which contains $\{g, a, e\}$. Now, suppose $(G, \{a, h, c, f, e\})$ and $(G, \{b, h, c, f, d\})$ are two instances of EXT IS. It is easy to show that $(G, \{a, h, c, f, e\})$ is a *yes*-instance of EXT IS, because $\{a, c, e\}$ is a maximal independent set. But there is no maximal independent set included in $\{b, h, c, f, d\}$, since at least one of vertices $\{a, e, g\}$ can be added to any independent set of $\{b, h, c, f, d\}$ and makes a bigger independent set, so $(G, \{b, h, c, f, d\})$ is a *no*-instance of EXT IS.

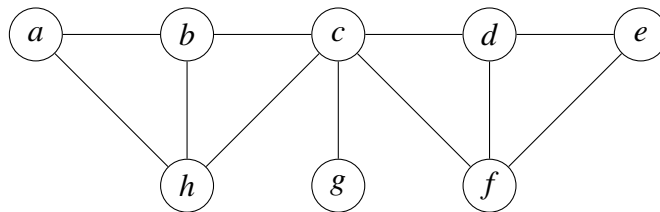


Figure 2.2: The graph $G = (V, E)$ as a part of the instances of EXT IS and EXT VC.

This type of decision problem related to an extension optimization problem is encountered in many efficient enumeration algorithms: when enumerating all minimal solutions for an instance of a problem, one usually asks for a given of pre-solution U , if there exists a *minimal* solution S with $U \subseteq S$. This decision problem has been considered in many studies in the framework of enumeration, for example, see [31, 32, 33, 59, 80, 92, 93, 95, 120, 119, 142, 152]. For instance, MINIMAL VERTEX COVER EXTENSION (MVCEXT for short) consists of a graph $G = (V, E)$ and a set $Y \subseteq V$ as an input, and the goal is to find all sets X enjoying these two properties: (1) $X \supseteq Y$, and (2) X is a minimal vertex cover in G . Let κ -MVCEXT be the same problem with the additional demand (3) $|X| \leq k$. These two problems have been studied in [59]. More generally, the question of finding extensions to minimal solutions was encountered in the context of proving hardness results for (efficient) enumeration algorithms for Boolean formulae, in the context of matroids and similar situations; see [32, 120].

Moreover, in [32], it is shown that extension of partial solutions is NP-hard for computing prime implicants of the dual of a Boolean function; a problem which can also be seen as trying to find a minimal hitting set for the prime implicants of the input function. Interpreted in this way, the proof from [32] yields NP-hardness for the minimal extension problem for 3-HITTING SET (but polynomial-time solvable if $|U|$ is constant). This result was extended in [17] to prove NP-hardness for computing the extensions of vertex subsets to minimal dominating sets (EXT DS), even restricted to planar cubic graphs. Similarly, it was shown in [16] that extensions to minimal vertex covers restricted to planar cubic graphs is NP-hard.

An *independent system* is a set system (V, \mathcal{E}) , $\mathcal{E} \subseteq 2^V$, that is hereditary under inclusion. The extension problem EXT IND SYS (also called FLASHLIGHT) for independent system was proposed in [131]. In this problem, given as input $X, Y \subseteq V$, one asks for the existence of a maximal independent set including X that does not intersect with Y . Lawler et al. proved that EXT IND SYS is NP-complete, even when $X = \emptyset$ [131]. In order to enumerate all (inclusion-wise) minimal dominating sets of a given graph, Kanté et al. studied a restriction of EXT IND SYS: finding a minimal dominating set containing X . They proved that EXT DS is NP-complete, even in special graph classes like split graphs, chordal graphs and line graphs [117, 116]. Moreover, they proposed a linear algorithm for split graphs when X, Y is a partition of the clique part [115].

The case where the forced set U satisfies some graph property like being a clique or an independent set has already been studied in some context:

- **Strong clique.** A *strong clique* in a graph is a clique intersecting every maximal independent set. In [107], Hujdurovic et al. have studied six algorithmic decision problems related to strong cliques. They proved that a clique C in a graph G is not strong if and only if it is dominated by an independent set $I \subseteq V(G) \setminus C$. In other words, they showed that a clique C is strong if and only if there is a minimal vertex cover of G extending C . Moreover, by introducing the extension variant of the strong clique problem, the authors made a generalized version of their problems. EXT STRONG CLIQUE consists of a graph G and a clique C in G , and it asks whether G contains a strong clique $C' \supseteq C$. A survey of Strong clique can be found in [M. Milanič, Strong cliques and strong stable sets in graphs, chapter to appear in: Topics in Algorithmic Graph Theory, Cambridge University Press, edited by L. Beineke, M. Golubic, R. Wilson]
- **Disjoint independent set.** The problem of deciding whether a graph has a pair of disjoint maximal independent sets is an old problem in graph theory which was introduced by Berge in early 70's. Moreover, recently, many studies have been done in order to find a pair of disjoint maximal/minimal subsets of graphs satisfying same property [53, 65, 150, 108]. For example in [105], it is shown that deciding whether a graph has two disjoint maximal independent sets is NP-complete problem. By assuming that one of the disjoint subsets is given in advance, the problem transforms to the extension decision problem. For example, suppose a graph $G = (V, E)$ together with a maximal independent set $x \subseteq V$ is given in advance and it is asked whether G has a maximal independent set included in $V \setminus X$.
- **Extension of partial representation.** In the context of extending a given partial representation into a full one for a subclass of intersection graphs, several results are proposed in

[6, 124, 122, 121, 123, 48]. For instance, in [124, 122], for interval graphs, the problem asks, for a given interval graph with some intervals pre-drawn by the input, whether the partial representation can be extended to a representation of the entire graph. In [122, 124], the authors gave a linear-time algorithm for respectively extending proper interval representations and interval graphs using PQ-trees. In [122], the authors also introduced a more general problem of bounded representations of unit interval graphs which is NP-complete, where the input constrains the positions of some intervals by lower and upper bounds.

2.3 Price of extension

Considering the possibility that some set U might not be extensible to any minimal or maximal solution, one might ask how far is U from an extensible set. One idea to evaluate this, is to ask how much U has to be altered when aiming for a minimal solution. Described differently for our extension problems at hand, we want to discuss how many elements of U have to be deleted (resp. how many elements of $X \setminus U$ have to be added where X is the universe) for an anti-hereditary (resp. hereditary) problem, in order to arrive at a *yes*-instance of the extension problem. In this regard, we define a new notion *price of extension* (PoE), in an attempt to understand what effect the additional pre-solution constraint has on the possibility of finding minimal (resp. maximal) solutions. A similar approach has already been used in the past under the name *the Price of Connectivity* in [38] for the context of connectivity because it is a crucial issue in networking applications; this notion has been introduced in [38] for MIN VC and is defined as the maximum ratio between the connected vertex cover number and the vertex cover number. In our context, the goal of PoE is to quantify how close efficiently computable extensible subsets of the given pre-solution U are to U or to the largest possible extensible subsets of U . To formally discuss this concept, we introduce the following optimization problems. From an instance (G, U) of an anti-hereditary (resp. a hereditary) problem Π , two new optimization problems denoted by $\text{EXT}_{\max}\Pi$ (resp. $\text{EXT}_{\min}\Pi$) and defined as follows:

$\text{EXT}_{\max}\Pi$

Input: An instance x together with a forced set U .

Solutions: Minimal solution S of $G = (V, E)$.

Output: Solution S that maximizes $|S \cap U|$.

$\text{EXT}_{\min}\Pi$

Input: An instance x together with a co-forbidden set U .

Solutions: Maximal solution S of x .

Output: Solution S that minimizes $|S \cup U|$.

To be clear, let us give an example.

Example 2.3. Suppose that (G, U) is an instance of $\text{Ext}_{\max}\text{VC}$ where $G = (V, E)$ is presented in Figure 2.3 and $U \subseteq V = \{c, d, e\}$. Easily, one can see that (G, U) is a *no*-instance of ExtVC while vertex sets $\{c, d\}, \{e\}$ are extendible to a minimal vertex cover. Since we are seeking a minimal vertex cover containing a maximum number of vertices in U , then the optimal solution is $\{b, c, d\}$. Now, consider (G, U) with $U = \{a, b\}$ as an instance of $\text{Ext}_{\min}\text{IS}$. Suppose S is a maximal independent set of G . If $a \in S$, by maximality $e \in S$ too, and if $b \in S$, then $\{c, d\} \subset S$. Since for $\text{Ext}_{\min}\text{IS}$, we are looking for a maximal independent set including a minimum number of vertices outside of U , then the optimal solution of $\text{Ext}_{\min}\text{IS}$ for (G, U) is $\{a, e\}$.

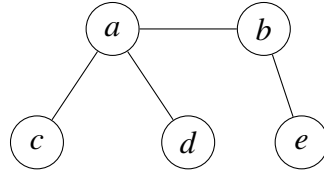


Figure 2.3: The graph $G = (V, E)$ as a part of the instances of $\text{Ext}_{\max}\text{VC}$ and $\text{Ext}_{\min}\text{IS}$.

For $S = \text{Ext}_{\max}\Pi$ or $\text{Ext}_{\min}\Pi$, we denote by $\text{opt}_S(x, U)$ the value of an optimal solution. Since for both of the problems $\text{opt}_S(x, U) = |U|$ if and only if (x, U) is a *yes*-instance of $\text{Ext}\Pi$, we deduce that for an anti-hereditary (resp. a hereditary) problem Π , $\text{Ext}_{\max}\Pi$ (resp. $\text{Ext}_{\min}\Pi$) is NP-hard as soon as $\text{Ext}\Pi$ is NP-complete.

It is easy to see that the optimal solutions of $\text{Ext}_{\max}\text{VC}$ and $\text{Ext}_{\min}\text{IS}$ are related similarly to the existing relation between $\alpha(G)$ and $\tau(G)$ (i.e. the vertex cover number). Hence for any graph $G = (V, E)$ of n vertices, we have $\text{opt}_{\text{Ext}_{\max}\text{VC}}(G, U) + \text{opt}_{\text{Ext}_{\min}\text{IS}}(G, V \setminus U) = n$.

The price of extension PoE is defined exactly as the ratio of approximation, i.e. $\frac{\text{apx}}{\text{opt}}$. We say that Π admits a polynomial ρ - PoE if for every instance (x, U) , we can compute a solution S of G in polynomial time which satisfies $\text{PoE}(S) \geq \rho$ for $\text{Ext}_{\max}\Pi$ (resp. $\text{PoE}(S) \leq \rho$ for $\text{Ext}_{\min}\Pi$). Formally, we define it as follows:

Definition 2.4. For an instance (x, U) of $\text{Ext}_{\max}\Pi$ and (resp. $\text{Ext}_{\min}\Pi$), the price of extension (PoE) of a minimal (resp. maximal) solution S of Π on x is defined by $\text{PoE}(S) := \frac{|S \cap U|}{\text{opt}_{\text{Ext}_{\max}\Pi}(x, U)}$ (resp. $\text{PoE}(S) := \frac{|S \cup U|}{\text{opt}_{\text{Ext}_{\min}\Pi}(x, U)}$).

In the next section, we will introduce a framework of *maximum-minimal* (max-min for short) and *minimum-maximal* (min-max for short) graph optimization problems and we will show the relation between a problem in this framework and $\text{Ext}_{\max}\Pi$ and $\text{Ext}_{\min}\Pi$.

2.4 Max-min and min-max optimization problems in graphs

In this Section a framework of *max-min*, *min-max* optimization problems in graphs adapted from Manlove's thesis [136] will be proposed, then we will clarify the relation between the two frameworks of extension problems and min-max or max-min optimization problems. We

finish this section by reviewing most of the interesting results related to max-min or min-max optimization problems.

In the following, we show how to obtain a *max-min* or *min-max* graph optimization problem Π' , from a source graph problem Π .

Definition 2.5. Let $\Pi = \langle I, F, opt, m \rangle$ be an hereditary or an anti-hereditary graph optimization problem. Then the new optimization problem is $\Pi' = \langle I, F', opt', m \rangle$, where for any instance $G \in I$:

- $F'(G) := \begin{cases} \mu(G) & \text{if } \Pi \text{ is an anti-hereditary problem} \\ \psi(G) & \text{if } \Pi \text{ is a hereditary problem} \end{cases}$
- $opt' := \begin{cases} \min & \text{if } opt = \max \\ \max & \text{if } opt = \min \end{cases}$

If $opt = \max$ then new problem Π' is a *min-max* problem also called *lower* Π and if $opt = \min$, Π' is a *max-min* problem also called *upper* Π . In fact, this is the framework proposed by Manlove in [136]. He systematically studied a lot of problems, considering many different partial orders in this framework, for example see [137, 112] for coloring and 12 problems on coverings and independence problems in graphs. Some examples of partial orderings that are not hereditary or anti-hereditary are for instance k -exchange for independence set [135] or the problems called Maximal subset sum with digraph constraints [97]. A series of other problems that are neither hereditary nor anti-hereditary can be found in [136] and will be discussed in Chapter 6. As an example, if Π is MIN VC (resp. MAX IS), then Π' looks for a minimal vertex cover (resp. maximal independent set) of maximum (resp. minimum) size. The two new problems defined formally as follows:

MAX-MIN VC (also called UPPER VC)

Input: A graph $G = (V, E)$.

Solution: A minimal vertex cover $S \subseteq V$.

Output: Maximize $|S|$.

MIN-MAX IS (also called LOWER IS)

Input: A graph $G = (V, E)$.

Solution: A maximal independent set $S \subseteq V$.

Output: Minimize $|S|$.

To be clear, consider the following example.

Example 2.6. Consider the graph proposed in Figure 2.4 as an instance for both problems MAX-MIN VC and MIN-MAX IS. There are 3 different minimal vertex covers of G , $\{a, e, f\}$, $\{a, d\}$ and $\{d, b, c\}$ and whereas in MAX-MIN VC, we seek a minimal vertex cover of maximum size, so the size of an optimal solution is 3. For MIN-MAX IS there are also 3 feasible solutions for G , $\{b, c, d\}$, $\{a, e, f\}$ and $\{b, c, e, f\}$, since in MIN-MAX IS, we seek a maximal independent set of minimum size, so the size of an optimal solution equals 3.

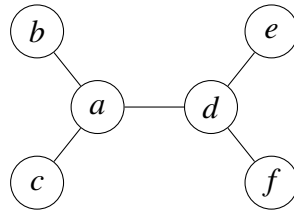


Figure 2.4: The graph G as an instance of MAX-MIN VC and MIN-MAX IS.

A particular case of $Ext_{min}\Pi$ (resp. $Ext_{max}\Pi$) for an instance (x, U) occurs when the pre-solution U is "empty-set" (resp. "whole-set"). So, if Π is an anti-hereditary problem, $Ext_{max}\Pi$ with (x, U) as an instance, is exactly equivalent to the well known problem MAX-MIN Π or UPPER Π when U is "whole-set", and if Π is a hereditary problem, $Ext_{min}\Pi$ with (x, U) as an instance is exactly equivalent to MIN-MAX Π or LOWER Π when U is "empty-set". Therefore, $Ext_{max}\Pi$ contains the MAX-MIN Π for an anti-hereditary problem Π and $Ext_{min}\Pi$ contains the MIN-MAX Π for a hereditary problem Π .

Suppose $G = (V, E)$ is the instance of two problems MIN VC and MAX IS. The goal of $Ext_{max}\text{VC}$ for the case that the partial solution $U = V$ is to find a minimal vertex cover containing the maximum number of vertices of V , which is a solution for MAX-MIN VC, while the goal of $Ext_{min}\text{IS}$ when $U = \emptyset$ is to find a maximal vertex cover containing the minimum number of vertices of V which is a solution for MIN-MAX IS.

Considering a max-min or min-max version of a problem by "flipping" the objective is not a new idea; in fact, such questions have been posed before for many classical optimisation problems. Some of the most well known examples include the MIN-MAX IS problem [34], the MIN-MAX VC problem [30, 163], the LAZY BUREAUCRAT problem which is a min-max version of SUBSET SUM problem which asks for a given set of integers w_1, \dots, w_n and a value W , does any subset of them sum to precisely W ? [86, 96], the MAX-MIN DS problem [2, 17], and the MIN-MAX MATCHING problem [162].

The MIN-MAX IS problem, also called *minimum independent dominating set* (MIN ISDS) asks, given a graph $G = (V, E)$, for a subset $S \subseteq V$ of minimum size that is simultaneously independent and dominating. From the NP-hardness and exact solvability point of views, MIN ISDS is equivalent to UPPER VC [137], but they seem to behave differently in terms of approximability and parameterized complexity [12]. Although MIN ISDS is polynomially solvable in strongly chordal graphs [77], it is hard to approximate within $n^{(\epsilon-1)}$, for any $\epsilon > 0$, in certain graph classes [77, 61]. Regarding parameterized complexity, Fernau [81] presented an fpt-algorithm for UPPER VC with running time $O^*(2^k)$, where k is the size of an optimum solution, while it is proved that MIN ISDS with respect to the standard parameter is W[2]-hard. Moreover, Boria et al. [30] provided a tight approximation result for UPPER VC in general graphs: they presented a $n^{\frac{1}{2}}$ approximation algorithm together with a proof that the UPPER VC problem is NP-hard to approximate within $n^{(\epsilon-\frac{1}{2})}$, for any $\epsilon > 0$. Furthermore, they presented a parameterized algorithm with running time (1.5397^k) where k is the standard parameter, by modifying the algorithm of [81]; they also showed that the weighted versions of UPPER VC and MIN ISDS are in FPT with

respect to the treewidth.

The MIN-MAX MATCHING problem was studied as early as 1980, when Yannakakis and Gavril showed that it is NP-hard even in some restricted cases [162]. MIN-MAX MATCHING problem is polynomially equivalent to the *minimum edge dominating set* (denoted by MIN EDS); the problem asks, given a graph $G = (V, E)$, for a subset $S \subseteq E$ of minimum size dominates all edges of G . An independent edge dominating set is an edge dominating set in which no two elements are adjacent. The MIN-MAX MATCHING problem is also known *minimum independent edge dominating set* (denoted by MIN ISEDS) and asks, given a graph $G = (V, E)$, for a subset $S \subseteq E$ of minimum size that is simultaneously independent and dominating. In 2006, it was proved that MIN-MAX MATCHING is NP-hard to approximate within a factor better than $\frac{7}{6}$ [50]. Later, in [75], the result improved, where the authors proved that the problem is hard to approximate with ratio better than 1.18, assuming $P \neq NP$. Finally, Dudyecz et al. proved that the MIN-MAX MATCHING is hard to approximate within ratio 2, assuming Unique Games Conjecture (UGC) [73].

2.5 Overview

In this thesis, we study a special type of optimization problems, called *extension problems*. We study some graph hereditary and graph anti-hereditary optimization problems in this framework from a computational complexity, (in-)approximability and parameterized complexity points of view. Informally, in the extension version of an anti-hereditary (resp. a hereditary) optimization problem Π , we consider an instance of Π with an additionally - specified set U which is called a pre-solution, while the goal is to obtain an inclusion-wise *minimal* (resp. maximal) solution S of optimum value such that $U \subseteq S$ (resp. $U \supseteq S$).

Concerning finding a minimal solution containing a forced set, with optimal value, we study the *weighted edge cover* problem in Chapter 3. The input of an extension version of weighted edge cover consists of both a weighted graph $G = (V, E, w)$ and a forced set $U \subseteq E$, and the task is to extend U to an inclusion-wise minimal edge cover of optimal value. First we show that for a given graph G and a forced set U , EXT EC is an NP-complete problem even for bipartite graphs of maximum degree 3. Next, we consider both maximization and minimization variants of the problem in complete graphs. For a general weight function, we show that MIN EXT WEC is not approximable at all, but MAX EXT WEC is 0.5 approximable. Moreover, we consider some other weight functions, and show that MIN EXT WEC when the weight function satisfies the c -relaxed (or extended c -relaxed) triangle inequality for $c > 1$ is in APX while when it satisfies the c -relaxed triangle (resp. extended c -relaxed) triangle inequality for $c \leq 1$ (resp. $c = 1$) it is polynomial-time solvable. Furthermore, we show that both optimization problems are polynomial-time solvable in graphs with bounded tree-width.

Further, we focus on a particular case of MAX EXT WEC when the forced set is empty. The problem is called WEIGHTED UPPER EC in which, for a given edge weighted graph, the aim is to find a minimal edge cover of maximum total weight. We study the problem for some graph classes from an (in-)approximability point of view: we prove that WEIGHTED UPPER EC in

complete graphs is equivalent to MAX STAR FOREST problem in general graphs. We also show that, WEIGHTED UPPER EC in bipartite graphs and split graphs with binary weights is as hard as MAX IS in general graphs. Moreover, we show that the problem in k -trees and in graphs of bounded degree is APX-complete.

We consider the extension variants of some vertex graph problems in Chapter 4. We study the decision problem EXT Π when Π is one of the following vertex graph problems: VERTEX COVER, INDEPENDENT SET, CONNECTED VERTEX COVER, NON SEPARATING INDEPENDENT SET and DOMINATING SET. For all of these problems we show that the problems are NP-complete for restricted cases of input graphs. Also, we find many graph classes where the problems are solvable in polynomial time. Furthermore, we study the parameterized complexity of these problems, with respect to the size of the pre-solution U as well as the dual set \bar{U} . All these complexity considerations are also carried out in very restricted scenarios, be it degree or topological restrictions (bipartite, planar or chordal graphs). We further introduce the *price of extension* (PoE), measuring the distance of U to the closest set that can be extended, which results in natural optimization problems related to extension problems for which we discuss polynomial-time approximability. Moreover, for some of the problems like CONNECTED VERTEX COVER, we focus on a particular case (G, \emptyset) of EXT_{max} CVC, which is also known with MAX-MIN CVC. We study the latter problem from an (in-)approximability point of view and present some interesting results.

In Chapter 5, we concentrate on some problems related to edges in graphs. Particularly, we consider the extension variant of the generalizations of EDGE COVER, EDGE DOMINATING SET and EDGE MATCHING, denoted by EXT r -EC, EXT r -EDS and EXT r -DCPS respectively. In the generalized version of edge cover (resp. edge dominating set) denoted by r -EDGE COVER (resp. r -EDGE DOMINATING SET), for a given graph $G = (V, E)$, $S \subseteq E$ is a feasible solution if $\forall v \in V$, v is covered by (resp. $\forall e \in E$, e is dominated by) at least r edges in S . In the generalized version of EDGE MATCHING denoted by r -DCPS, for a given graph $G = (V, E)$, $S \subseteq E$ is a feasible solution if $\forall v \in V$, v is incident to at most r vertices of S .

We show all of EXT r -EC, EXT r -EDS and EXT r -DCPS are NP-complete for graphs of maximum degree $r+2$, moreover we prove for the particular case $r = 1$, all the mentioned problems are NP-complete for subcubic planar bipartite instances. We also study the parameterized complexity of these problems, with respect to the size of pre-solution U or its dual set \bar{U} , which leads to some positive parameterized results. Furthermore, we discuss the concept *price of extension* (PoE) for all the mentioned problems, and establish some hardness results in terms of approximability for corresponding natural optimization problems.

Finally, in Chapter 6, we introduce a general partial-order based framework to model extension for monotone problems with the attempt to highlight the unified structure of such types of problems that seem to appear in many different scenarios. Informally, in the extension version of an optimization problem Π , we consider an instance of Π with additionally a pre-solution and a partial order on those as an input, while the goal is to *extend* the pre-solution to a *minimal* one of optimum value. This partial order \leq reflects not only the notion of *extension* but also that of *minimality* such that, for a pre-solution U and a solution S , S *extends* U if $U \leq S$. A solution S is *minimal* if there exists no solution $S' \neq S$ with $S' \leq S$. Admittedly, our framework does

not cover all problems of this flavor. We also add some hardness results for problems like BIN PACKING, FEEDBACK VERTEX SET and FEEDBACK EDGE SET.

Extended Weighted Edge Cover Problem

Contents

| | | |
|------------|---|-----------|
| 3.1 | Introduction | 62 |
| 3.1.1 | Some general definitions | 64 |
| 3.1.2 | Related work | 65 |
| 3.1.3 | Contributions | 67 |
| 3.2 | Extended weighted edge cover in general graphs | 67 |
| 3.3 | MIN EXT WEC in complete graphs | 73 |
| 3.4 | MAX EXT WEC in complete graphs | 81 |
| 3.5 | Weighted upper edge cover | 84 |
| 3.5.1 | Weighted Upper EC in complete graphs | 85 |
| 3.5.2 | Weighted Upper EC in bipartite graphs | 87 |
| 3.5.3 | Weighted Upper EC in split graphs | 89 |
| 3.5.4 | Weighted Upper EC in k -trees | 91 |
| 3.5.5 | Weighted Upper EC in bounded degree graphs | 97 |
| 3.6 | Conclusion | 98 |

Most of the results presented in this chapter have appeared in the following conference and/or journal papers:

- Weighted Upper Edge Cover: Complexity and Approximability. In Proceedings of the 13th International Workshop on Algorithms and Computation, WALCOM 2019. Joint work with Kaveh Khoshkhah, Jerome Monnot and Florian Sikora.
- Extended spanning star forest problems. In Proceedings of the 11th Annual International Conference on Combinatorial Optimization and Applications, COCOA 2017. Joint work with Kaveh Khoshkhah, Jerome Monnot and Dirk Oliver Theis.

- Complexity and approximability of extended Spanning Star Forest problems in general and complete graphs. Journal of Theoretical Computer Science, 2019. Joint work with Kaveh Khoshkhan, Jerome Monnot and Dirk Oliver Theis.
- Weighted Upper Edge Cover: Complexity and Approximability. Journal of Graph Algorithms and Applications (submitted, invited to the special issue of WALCOM 2019), Joint work with Kaveh Khoshkhan, Jerome Monnot and Florian Sikora.

3.1 Introduction

For a given graph $G = (V, E)$, a set $S \subseteq E$ is called an *edge cover* if and only if S covers all vertices of V , i.e. each $v \in V$ is incident to at least one $e \in S$. The goal of the optimization version of the problem is to find an edge cover of minimum size. The weighted variant of edge cover consists of an edge weighted graph $G = (V, E, w)$ as an input and the goal is to find an edge cover S with optimal value, i.e. minimizing $w(S) = \sum_{e \in S} w(e)$. In this chapter, we study the weighted version of edge cover in the framework of extension problems.

The input of an extension version of weighted edge cover consists of both a weighted graph $G = (V, E, w)$ and a forced set $U \subseteq E$, and the task is to extend U to an inclusion-wise minimal edge cover of optimal value. Both maximization and minimization variants of the problem are studied in this chapter. We study the two problems in computational complexity and (in)-approximability points of view. The minimization (resp. maximization) variant is denoted by MIN EXT WEC (resp. MAX EXT WEC) and is formally defined as follows:

MAX EXT WEC

Input: A weighted graph $G(V, E, w)$ with a forced set $U \subseteq E$ where $w(e) \geq 0$ for $e \in E$.

Solution: A minimal edge cover $S \subseteq E$ such that $U \subseteq S$.

Output: Maximizing $w(S) = \sum_{e \in S} w(e)$.

MIN EXT WEC

Input: A weighted graph $G(V, E, w)$ with a forced set $U \subseteq E$ where $w(e) \geq 0$ for $e \in E$.

Solution: A minimal edge cover $S \subseteq E$ such that $U \subseteq S$.

Output: Minimizing $w(S) = \sum_{e \in S} w(e)$.

By minimality, each minimal edge cover does not contain any P_4 or C_3 as a subgraph (induced), thus, any minimal edge cover is a collection of stars. Hence, for an instance (G, U) if the forced set U contains a P_4 or C_3 as an induced subgraph, easily one can deduce that there do not exist any minimal edge cover of G which contains U . Therefore, along this chapter, we assume that the forced set U is a packing of stars and the goal of MAX EXT WEC (resp. MIN EXT WEC) is to find a collection of non-trivial stars¹, maximizing (resp. minimizing) its weight.

Notice also that a minimal edge cover containing a given forced set may not exist even if U is just a matching. For example consider a P_4 where the forced set U is the middle edge of the

¹A star is non-trivial if it forms a $K_{1,\ell}$ with $\ell \geq 1$.

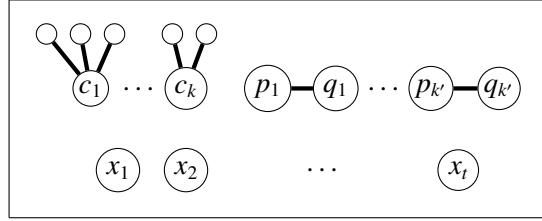


Figure 3.1: Bold edges corresponds to forced edges of U : sets S_U containing stars with at least 2 leaves and M_U containing stars with one leaf, are indicated on the left side and on the right side of the figure respectively. Set $X = \{x_1, \dots, x_t\}$ is the vertices of the graph that are outside of U .

path. To cover both endpoints of the path, two pendant edges must be in any edge cover, although, adding these edges contradicts to the minimality. However, for complete graphs when the forced set U is a collection of non-trivial stars, there always exists some minimal edge cover containing U . From now, we assume that the forced set $U = M_U \cup S_U$ is decomposed into a matching $M_U = \{p_i q_i : i = 1, \dots, k'\}$ of k' edges and a set $S_U = \{F_1, \dots, F_k\}$ of k vertex-disjoint stars with at least two leaves (c_i will be the center of F_i and $C = \{c_1, \dots, c_k\}$ is the set of centers in S_U). An illustration of these definitions is depicted in Figure 3.1.

MIN EXT WEC when $U = \emptyset$ is equivalent to MIN WEC and it can be described as follows: Given a weighted graph $G = (V, E, w)$, the MIN WEC problem consists of finding an edge cover $S \subseteq E$ of G minimizing $w(S) = \sum_{e \in S} w(e)$. This problem is polynomial-time solvable (see chapters 33 and 34 volume A of [151]). MAX EXT WEC, when the forced set $U = \emptyset$, is called WEIGHTED UPPER EC. For a given weighted graph $G = (V, E, w)$ as an instance of WEIGHTED UPPER EC, the goal is to find a minimal edge cover $S \subseteq E$ of G , maximizing $w(S) = \sum_{e \in S} w(e)$. We study this problem for different graph classes in the framework of (in)-approximability, in Section 3.5. By minimality, any minimal edge cover is a collection of non-trivial stars. MAX EXT WEC when the forced set $U = \emptyset$ and trivial stars (isolated vertices) are allowed to be in feasible solutions is known as MAX STAR FOREST problem. The former problem has been investigated intensively recently for unweighted and weighted graphs. The problem is defined formally as follows:

MAX STAR FOREST

Input: A weighted graph (G, w) where $G = (V, E)$ and $w(e) \geq 0$ for $e \in E$.

Solution: A spanning star forest $S = \{S_1, \dots, S_p\} \subseteq E$.

Output: Maximizing $w(S) = \sum_{e \in S} w(e) = \sum_{i=1}^p \sum_{e \in S_i} w(e)$.

In Section 3.5, we attempt to show the relation between WEIGHTED UPPER EC problem and MAX STAR FOREST problem in (in)-approximability point of view. In other words, we try to understand how much MAX STAR FOREST problem can be intractable when having trivial stars in feasible solutions are not allowed.

3.1.1 Some general definitions

Residue approximation ratio

For algorithms finding an optimal solution containing a pre-solution U , it often does not matter whether we are optimizing the weight of the overall solution S or the additional part $S \setminus U$, which is called *residue*. Nevertheless, in approximation point of view, this difference can be significant in a way that an algorithm which is producing an approximation residue can be more intractable than one is producing an approximation containing the given pre-solution. This motivated Weller et al. to propose a new definition of approximability for extension problems [161].

Definition 3.1 ([161]). *Let x be an instance of an extension optimization problem Π . Let R and U be such that $R \cup U$ is a solution of Π for x . Then, we call R a residue for (x, U) . Let R^* be a residue of optimal value for (x, U) , then the residue approximation ratio is defined by $\rho_r := \frac{m(R)}{m(R^*)}$ where m is the measure function.*

By Definition 3.1, Weller et al. mentioned that an extension problem Π belongs to RAPX if and only if it admits a polynomial algorithm with constant residue approximation ratio [161]. They also noted that $\text{RAPX} = \text{APX}$ for the special case $R = \emptyset$, so generally for any arbitrary set R , $\text{RAPX} \subseteq \text{APX}$. They also showed that the extension variant of VERTEX COVER is in $(\text{RAPX} \cap \text{APX})$, while the extension variant of BIN PACKING is in $\text{APX} \subseteq \text{RAPX}$. The input of BIN PACKING consists of a set $X = \{x_1, \dots, x_n\}$ of items and a weight function w that associates rational numbers $w(x_i) \in (0, 1)$ to items. A feasible solution is a partition π of X such that, for each set $Y \in \pi$, $\sum_{y \in Y} w(y) \leq 1$. The traditional aim is to find a feasible π such that $|\pi|$ is minimized.

Cost Function Variants

In this chapter, we consider different variants of the problem according to the weight (or cost) function $w : E \rightarrow \mathbb{R}$ for some different classes of graph $G = (V, E, w)$. The general version assumes that w is any non-negative integer weight function. We also make some variations of the problem, by considering *univariate weights*, *bivariate weights* and *trivariate weights* corresponding to the case $w(e) \in \{a\}$ with $0 \leq a$, $w(e) \in \{a, b\}$ with $0 \leq a < b$ and $w(e) \in \{a, b, c\}$ where $0 \leq a < b < c$ are 3 reals respectively. The particular cases $(a = 1)$ and $(a = 0, b = 1)$ are more interesting and called here *unary weights* and *binary weights*, respectively. The MIN EXT WEC (resp. MAX EXT WEC) for the case of binary weights is denoted by MIN $\{0, 1\}$ -EXT WEC (resp. MAX $\{0, 1\}$ -EXT WEC). It is easy to see that MIN EXT WEC (resp. MAX EXT WEC) in general graphs with unary weights is equivalent to MIN EXT WEC (resp. MAX EXT WEC) in complete graphs with binary weights; keep all the edges of the general graphs in the complete graph with weight 1 and add all nonexistence edges with weight 0. In these cases, the size of a minimal edge cover is the number of leaves in all its components, and the goal in this case is to find a packing of stars with minimum (resp. maximum) number of leaves.

In addition, in Section 3.3, another version of weight function w is considered: w satisfies the *c-relaxed triangle inequality*. Mainly consider that the *c-relaxed triangle inequality* might be satisfied outside the subgraph induced by $V(U)$ where $V(U) = \{u, v : uv \in U\}$, i.e. inside $V \setminus V(U)$ because the structure of feasible solutions are strongly constrained by subset U .

Definition 3.2 (*c*-relaxed triangle inequality). For a fixed $c > 1/2$, a weight function w on K_n satisfies the *c*-relaxed triangle inequality, if:

$$\forall x, y, z \in V(K_n), \quad w(x, y) \leq c(w(x, z) + w(z, y)) \quad (3.1)$$

The case $c = 1$ is usually called in the literature *triangle inequality* while for $c \in (1/2; 1)$ it is called *sharpened triangle inequality*. Note that the extreme case $c = 1/2$ becomes trivial since all edges must have the same weight. A detailed motivation of the study of the TRAVELING SALESMAN PROBLEM satisfying sharpened triangle inequalities is given in [26]. In the context of extended problems, Definition 3.2 leads to a new definition called the *Extended c-relaxed triangle inequality*:

Definition 3.3 (extended *c*-relaxed triangle inequality). For a fixed $c \geq 1$, a weight function w on K_n satisfies the extended *c*-relaxed triangle inequality, if:

- (i) $w(e) = 0$ for $e \in U$;
- (ii) for all $\{x, y, z\} \not\subseteq V(U)$, w satisfies the *c*-relaxed triangle inequality.

Condition (i) of Definition 3.3 refers to the discussion in [161] which argues regarding the fixed forced set U in any feasible solution S , the “residue” part of S (i.e. the part given in $S \setminus U$), is important to value. Another consequence of conditions (i) and (ii) concerns the valuation of w restricted to the subgraph induced by $V(U)$ (except for edges of U): this function does not satisfy any specified property. The main reason is that they could never contribute in any minimal edge cover containing U . Finally, the reason for assuming $c \geq 1$ is that condition (ii) implies $\max\{w(xz), w(yz)\} \leq c \min\{w(xz), w(yz)\}$ when $xy \in U$ and $z \notin V(U)$.

It is well known that an optimization problem is easier to approximate when the input is a complete weighted graph satisfying the *triangle inequality*, for example in the TRAVELING SALESMAN PROBLEM [52]. In Section 3.5, we make a new variant of the problem by introducing a generalization of this notion which works for any class of graphs.

Definition 3.4. An edge weighted graph $G = (V, E, w)$ satisfies the cycle inequality, if for every cycle C in G we have:

$$\forall e \in C, \quad 2w(e) \leq w(C) = \sum_{e' \in C} w(e').$$

Clearly, for complete graphs, cycle and triangle inequality notions coincide. Definition 3.4 is interesting when focusing on classes of graphs which are close to complete graphs, like split graphs or k -trees.

3.1.2 Related work

For a given edge weighted graph, finding an edge cover of minimum value can be computed in polynomial time (see Chapters 19 and 27 in [151]), but, to the best of our knowledge, the extended version of *edge cover* has not been studied in the literature.

UPPER EC problem is closely-related to MAX STAR FOREST problem and *minimum dominating set problem* that have been investigated intensively in unweighted graphs. MAX STAR FOREST problem looks for a collection of vertex disjoint stars with maximum number of leaves. The *minimum dominating set problem* (denoted MIN DS) seeks the smallest dominating set of G . Let $msf(G)$ and $\gamma(G)$ denote the size of optimal solution of MAX STAR FOREST and MIN DS for an unweighted graph G (without isolated vertices) of order n , respectively, we have $msf(G) = np - \gamma(G)$. On the other hand, UPPER EC problem in unweighted graph was firstly investigated in [137], where it is proven that the problem is equivalent to MIN DS. Let $uec(G)$ denote the size of optimal solution of UPPER EC for an unweighted graph G (without isolated vertices) of order n , respectively, we have $uec(G) = n - \gamma(G)$. Thus, using the complexity results known on MIN DS, we deduce that UPPER EC are NP-hard in planar graphs of maximum degree 3 [88], chordal graphs [29] (even in *undirected path graphs*, the class of vertex intersection graphs of a collection of paths in a tree), bipartite graphs, split graphs [22] and k -trees with arbitrary k [56], and they are polynomial solvable in k -trees with fixed k , convex bipartite graphs [60], strongly chordal graphs [77]. Concerning the approximability, an APX-hardness proof with explicit inapproximability bound as well as a combinatorial 0.6-approximation algorithm for UPPER EC problem proposed in [143]. Better algorithms with approximation ratio 0.71 and 0.803 are given respectively in [49] and [11]. For any $\varepsilon > 0$, UPPER EC is hard to approximate within a factor of $\frac{259}{260} + \varepsilon$ unless $P = NP$, while it admits a PTAS in k -trees (with arbitrary fixed k) [143]. A graph of order n is called c -dense if it contains at least $c\frac{n^2}{2}$ edges and it is called *everywhere- c -dense* if the minimum degree is cn ; when $c \in (0; 1)$ is a constant, we say *dense* and *everywhere-dense* graphs [10]. In [104], it is proved that UPPER EC remains APX-complete in c -dense graphs; they proposed an approximation algorithm with ratio $0.804 + 0.196\sqrt{c}$ for c -dense graphs, while proving that the problem does not admit a PTAS assuming $P \neq NP$. He and Liang in [103] claimed that UPPER EC on everywhere- c -dense graphs are easier than c -dense graph; they proved the claim by proving that for a given graph of order n and minimum degree $\delta(n)$ for every vertex, UPPER EC is APX-complete when $1 \leq \delta(n) \leq O(1)$, is NP-hard but allows a PTAS when $\omega(1) \leq \delta(n) \leq O(n^{1-\epsilon})$ for some constant ϵ , and is not NP-hard assuming ETH when $\delta(n) \geq \omega(n^{1-\epsilon})$ for every constant $\epsilon > 0$.

In contrast, to the best of our knowledge, for edge weighted graphs with non-negative weights, no result for WEIGHTED UPPER EC is known, although some results are given for MAX STAR FOREST problem: a 0.5-approximation is given in [143] (which is the best ratio obtained so far) and polynomial-time algorithms for special classes of graphs such as trees and cactus graphs are presented in [143, 144]. Negative approximation results are presented in [143, 44, 49]. For any $\varepsilon > 0$, the edge weighted version is NP-hard to approximate within $\frac{10}{11} + \varepsilon$ [44]. A subset of vertices V' is called *non-blocking* if every vertex in V' has at least one neighbor in $V \setminus V'$. Actually, *non-blocking* is dual of dominating set and vice versa. For a given graph $G = (V, E)$ and a positive integer k , the NON-BLOCKER problem asks if there is a *non-blocking* set $V' \subseteq V$ with $|V'| \geq k$. Hence, for unweighted graphs, optimal values of NON-BLOCKER and UPPER EC are equal. In [63], Dehne et al. propose a parameterized perspective of the NON-BLOCKER problem. They give a linear kernel and an fpt-algorithm running in time $O^*(2.5154^k)$. They also give faster algorithms for planar and bipartite graphs.

3.1.3 Contributions

In this chapter, we investigated the extension variant of weighted edge cover problem. In Section 3.2, we show that the decision variant of the problem is NP-hard in general graphs, while we show that the problem is solvable in polynomial time in graphs with bounded tree width. In Section 3.3 and Section 3.4, the optimization variants of the problem is studied in weighted complete graphs with considering different weight functions. Particularly, in Section 3.3, we will show that the MIN EXT WEC problem for general weight function w is not approximable at all, while the problem is APX-complete when w satisfies c -relaxed (or extended c -relaxed) triangle inequality for $c > 1$. Further, we will propose a polynomial time algorithm for the MIN EXT WEC problem when the weight function w satisfies c -relaxed (resp. extended c -relaxed) triangle inequality for $c \leq 1$ (resp. $c = 1$). The MAX EXT WEC will be considered from the (in-)approximability point of view in Section 3.4: we will show that there is no approximation algorithm with ratio better than $\frac{7}{8}$ for general weight function, while we propose a simple polynomial algorithm with a approximation ratio $\frac{1}{2}$.

Furthermore, in Section 3.5, we will study (in-)approximability of the MAX EXT WEC problem in a particular case, when the forced set $U = \emptyset$. We first will show that WEIGHTED UPPER EC in complete graphs is equivalent for its approximation to MAX STAR FOREST problem in general graphs. Moreover, we will show that the WEIGHTED UPPER EC problem is hard to approximate in bipartite and split graphs by giving an approximation preserving reduction from MAX IS. Further, we will prove the problem is APX-complete in k -trees with a constant k as well as in graph with bounded degree.

3.2 Extended weighted edge cover in general graphs

In this section we show that for a given graph $G = (V, E)$ and a packing of stars $U \subseteq E$, even deciding the existence of a minimal edge cover containing U is hard. However, we will show that when the input restricted to graphs with bounded tree-width, not only the decision version of the problem is decidable in polynomial time but also both MIN EXT WEC and MAX EXT WEC problems are polynomial-time solvable. The decision version of the problem denoted by EXT EC defined as follows.

EXT EC

Input: A graph $G = (V, E)$ and a packing of non-trivial stars $U \subseteq E$.

Question: Does G admit a minimal edge cover containing U ?

As a first result in this section, we show that EXT EC is NP-hard, even if the forced set U induces a matching.

Theorem 3.5. *EXT EC is NP-complete for general graphs even if U induces a matching in G .*

Proof. Clearly, EXT EC is in NP. For, given a graph $G = (V, E)$, a subset $U \subseteq E$, and a set $S \subseteq E$, it is straightforward to verify in polynomial time that S is a minimal edge cover of G

containing U . To show NP-hardness, we propose a simple reduction from SAT to the EXT EC. As it is already mentioned in Section 1.2, SAT is an NP-complete problem which consists of deciding if an instance $I = (\mathcal{C}, \mathcal{X})$ of SAT is satisfiable. Here, $\mathcal{C} = \{c_1, \dots, c_m\}$ and $\mathcal{X} = \{x_1, \dots, x_n\}$ are the set of clauses and variables respectively; a variable x_i which appears negatively will be denoted $\neg x_i$. From $I = (\mathcal{C}, \mathcal{X})$, we build an instance $I' = (G, U)$ of EXT EC as follows:

- $V(G) = V(C) \cup V(X)$ where $V(C) = \{w_j : c_j \in \mathcal{C}\}$ and $V(X) = \{v_i^0, v_i^1 : i = 1, \dots, n\}$,
- $U = \{v_i^0 v_i^1 : x_i \in \mathcal{X}\}$,
- $M = \bigcup_{j=1}^m (\{w_j v_i^1 : x_i \in c_j\} \cup \{w_j v_i^0 : \neg x_i \in c_j\})$,
- $E(G) = U \cup M$.

It is clear that I' is built in polynomial-time.

Suppose that I is satisfiable and let T be a satisfying truth assignment of I . For each clause c_j , we define $f(j)$ as an index of a variable $x_{f(j)}$ which satisfies clause c_j in T , we build a minimal edge cover S containing U as follows: $S = \{v_{f(j)}^1 w_j : c_j \in \mathcal{C} \wedge T(x_{f(j)}) = \text{true}\} \cup \{v_{f(j)}^0 w_j : c_j \in \mathcal{C} \wedge T(x_{f(j)}) = \text{false}\} \cup U$.

Conversely let S be a minimal edge cover containing U . Define an initial truth assignment T by setting $T(x_i) = \text{true}$ for all $x_i \in \mathcal{X}$. For each j ($1 \leq j \leq m$), by minimality w_j is incident to exactly one edge of S , since $S \subseteq U$ - let this edge be $w_j v_i^r$. If $r = 0$, set $T(x_i) = \text{false}$. T is a well-defined truth assignment, since by minimality of S , and since $S \subseteq U$, it is impossible that $v_i^0 w_j \in S$ and $v_i^1 w_j \in S$ for any $x_i \in \mathcal{X}$. Clearly T is a satisfying truth assignment for I . \square

From the result of Theorem 3.5, we can easily derive:

Corollary 3.6. *EXT EC is NP-complete for general graphs of maximum degree 3 and even if the packing of forced set U induces a matching.*

Proof. Instead of SAT, we start the reduction from a particular case of SAT, known as 2-BALANCED 3-SAT, denoted by (3, B2)-SAT which is already explained in Chapter 1. An instance $I = (\mathcal{C}, \mathcal{X})$ of (3, B2)-SAT is given by a set \mathcal{C} of CNF clauses defined over a set \mathcal{X} of Boolean variables such that each clause has exactly 3 literals, and each variable appears exactly 4 times in \mathcal{C} , twice negated and twice unnegated. Deciding whether an instance of (3, B2)-SAT is satisfiable is NP-complete [21, Theorem 1]. The reduction and proof remains the same as in Theorem 3.5 \square

As a point, in Chapter 5 (Theorem 5.2), we will prove that EXT EC is NP-complete for planar bipartite graphs of maximum degree 3.

Considering Corollary 3.6, we should focus on some instances which has always some feasible solutions for both MIN EXT WEC and MAX EXT WEC problems. Here, we propose a polynomial algorithm using dynamic programming for the two problems for bounded tree-width graphs.

Instead of operating an algorithm on the input graph, we operate it on the tree decomposition of the original graph. A short description of tree decomposition is proposed in Section 1.1.3.

As it is mentioned before, any minimal edge cover is a packing of non-trivial stars. To keep things simpler, we use the terminology of *spanning star forest* in our dynamic programming method. A spanning star forest of a given graph $G = (V, E)$, is a partition of V into some disjoint non-trivial stars. We distinguish among the cases that a vertex is labeled by *leaf*, *center* and *isolated*. Furthermore, we should pay attention to the cases that an edge belongs to the forced set U . By considering above two points, it is not hard to design a dynamic programming algorithm that solves MIN EXT WEC in time $O^*(c^{tw})$ for an instance (G, U) , where G is an edge weighted graph with a tree decomposition of width tw and c is a constant. We start our method by some definitions as follows.

Definition 3.7. For any vertex $v \in V$ of a given graph $G = (V, E)$ and a spanning star forest f , we assign a labeling of v as follows:

$$l_f(v, G) := \begin{cases} \text{center} & \text{if } v \text{ is a center of a star in } f, \\ \text{leaf} & \text{if } v \text{ is a leaf of a star in } f, \\ \text{isolated} & \text{if } v \text{ is a single vertex in } f. \end{cases}$$

Since both endpoints of each $K_{1,1}$ in the spanning star forest f can be center and leaf, one of them is labeled in arbitrary with *center* and the other with *leaf*. In the following, we propose a dynamic programming in order to find a spanning star forest containing the forced set with minimum total value.

Definition 3.8. For a weighted graph $G = (V, E, w)$, packing of stars U and a sub-labeling function l from $S \subseteq V$ to $\{\text{center}, \text{leaf}, \text{isolated}\}$, the value of minimum extended spanning star forest of G such that satisfies l and is allowed to have isolated vertices in S is defined as follows:

$$F_S^l(G, U) := \min_{f \in ESSF_l(G, U)} \{w(f) \mid \forall s \in S, l_f(s, G) = l(s) \text{ and } \forall s \notin S, l_f(s, G) \neq \text{isolated}\} \cup \{\infty\}. \quad (3.2)$$

In order to decrease the possible transactions between bags, we use a nice tree decomposition of G which is already explained in Section 1.1.3. Let T be a nice tree decomposition of G by rooted r . We note that, B_x and V_x are vertices appearing in node x and vertices appearing in the subtree rooted at x respectively. Actually, a minimum spanning star forest of G containing U is obtained from $\min_{l \in L} F_r^l(G, U)$ where L is a family of sub-labeling on B_r such that no vertex in B_r is labeled with *isolated*. The computation progresses from leaves of T to the root r by a postorder traversal and all valid sub-labelings of x are achieved by the possible sub-labelings of it's children. For each node x of T , let $G_x = G[V_x]$ and $U_x = G_x[E(G_x) \cap E(U)]$. We need to calculate $F_x^l(G_x, U_x)$ for all valid sub-labelings l on B_x for each node x in T . Since, there are four different types of nodes in T , we propose four types of function $F_x^l(G_x, U_x)$ as follows:

■ Leaf node

Suppose that $B_x = \{u\}$ is a leaf in T . It is clear that $|V_x| = 1$, $G_x = u$ and the only valid sub-labeling is $l(u) = \textit{isolated}$, so we have:

$$F_x^l(G_x, U_x) = \begin{cases} 0 & \text{if } l(u) = \textit{isolated}, \\ \infty & \text{otherwise.} \end{cases}$$

■ Introduce node

Suppose that y is the only child of x in T and $B_x = B_y \cup \{a\}$. The important thing here is the label of node a for all valid sub-labelings on B_x . Node a can accept all the three labels *isolated*, *leaf* and *center*:

- if $l(a) = \textit{isolated}$ and node a is not in $V(U_x)$, then by keeping all valid sub-labelings on B_y and adding a label "*isolated*" for node a we make some valid sub-labeling on B_x .
- if $l(a) = \textit{leaf}$, it means that node a is a leaf of a star s in $G[V_x]$, so by considering the structure of tree decomposition, center of s has to be in B_y . Every node $b \in B_y$ such that $ab \in E$ and b has one of the labels $\{\textit{isolated}, \textit{center}\}$ on B_y can be a candidate for the center of s .
- if $l(a) = \textit{center}$, it means that node a is a center of a star s in $G[V_x]$, so by considering the structure of tree decomposition, all of its leaves must be appeared in B_y . So every subset M of vertices with label *isolated* on B_y such that $M \subseteq N_G(a)$ can be a candidate for leaves of s .

In addition, we have to consider all above cases that whether node $a \in V(U_x)$ or not. Thus based on the label of a in the valid sub-labeling l on B_x , we consider following cases:

□ if $l(a) = \textit{isolated}$,

$$F_x^l(G_x, U_x) := \begin{cases} F_y^{l'}(G_y, U_y) & \text{if } a \notin V(U_x) \\ \infty & \text{otherwise.} \end{cases}$$

where $l'(v) = l(v)$ for each $v \in B_y$.

□ if $l(a) = \textit{leaf}$,

$$F_x^l(G_x, U_x) := \begin{cases} \min_{l(b)=\textit{center}, ab \in E(G_x)} \{F_y^{l'_b}(G_y, U_y) + w(ab), F_y^{l''_b}(G_y, U_y) + w(ab)\} & \text{if } a \notin V(U_x) \\ \min_{l(b)=\textit{center}, ab \in E(U_x)} \{F_y^{l'_b}(G_y, U_y) + w(ab), F_y^{l''_b}(G_y, U_y) + w(ab)\} & \text{if } d_{U_x}(a) = 1 \\ \infty & \text{if } d_{U_x}(a) > 1 \end{cases}$$

where $l'_b(v) := l(v)$ for $v \in B_y \setminus \{b\}$ and is *isolated* when $v = b$ and $l''_b(v) := l(v)$ for $v \in B_y \setminus \{b\}$ and is *center* when $v = b$.

□ if $l(a) = \textit{center}$, let $S = \{v \in V_x : l(v) = \textit{leaf} \text{ and } va \in E\}$,

$$F_x^l(G_x, U_x) := \begin{cases} \infty & \text{if } S = \emptyset, \\ \infty & \text{if } \exists b \in U_x : ab \in E(U_x) \\ & \text{and } d_{U_x}(b) > 1 \\ \min_{\emptyset \neq M \subseteq S, N_{U_x}(a) \subseteq M} \{F_y^{l^M}(G_y, U_y) + \sum_{v \in M} w(av)\} & \text{otherwise.} \end{cases}$$

where $l'_M(v) := \textit{isolated}$ for every $v \in M$ and equals $l(v)$ if $v \in B_y \setminus M$.

■ Forget node

Suppose that y is the only child of x in T and $B_y = B_x \cup \{a\}$ for $a \notin B_x$. Because of the structure of tree decomposition, we will not see again node a in other bags in the following of the algorithm. On the other hand, notice that the final solution does not contain any non-trivial stars (an isolated vertex). Hence, label a can not be *isolated* and all valid sub-labelings on B_y with $l(a) = \textit{isolated}$ will be invalid on B_x . Therefore, for each valid sub-labeling l on B_x we have:

$$F_x^l(G_x, U_x) := \min\{F_y^{l'}(G_y, U_y), F_y^{l''}(G_y, U_y)\}$$

where $l'(v) = l''(v) = l(v)$ for $v \in B_x$ and $l'(a) = \textit{center}$ and $l''(a) = \textit{leaf}$.

■ Join node

In the last case, suppose that y_1 and y_2 are children of x in T and $B_x = B_{y_1} = B_{y_2}$. Since, in the Join node x , two subgraphs V_{y_1} and V_{y_2} will be merged then the two sub-labelings l_1, l_2 defined on B_{y_1} and B_{y_2} respectively, have to be adapted to each other. For example, for a node $v \in B_x$, with $l_1(v) = \textit{leaf}$ and $l_2(v) = \textit{leaf}$ does not lead to a valid sub-labeling for B_x , because it makes a P_3 in G_x . Thus, for each valid sub-labeling l on B_x we have:

$$F_x^l(G_x, U_x) := \min_{\langle l_1, l_2 \rangle \in L} \{F_{y_1}^{l_1}(G_{y_1}, U_{y_1}) + F_{y_2}^{l_2}(G_{y_2}, U_{y_2})\}$$

$$\langle l_1, l_2 \rangle \in L \text{ iff } \begin{cases} \{l_1(v), l_2(v)\} = \{\textit{isolated}, \textit{leaf}\} & \text{if } l(v) = \textit{leaf}, \\ \{l_1(v), l_2(v)\} = \{\textit{isolated}, \textit{center}\} \text{ or } \{\textit{center}\} & \text{if } l(v) = \textit{center}, \\ \{l_1(v), l_2(v)\} = \{\textit{isolated}\} & \text{if } l(v) = \textit{isolated}. \end{cases}$$

Theorem 3.9. *MIN EXT WEC is solvable in polynomial time for bounded tree-width graphs.*

Proof. Now, we show that our proposed algorithm gives an optimal solution for MIN EXT WEC for a given weighted graph $G = (V, E, w)$ and a packing of non-trivial stars U as an instance. Actually, the output of the algorithm is a minimum spanning star forest of G contains U . We focus on recursive functions F and show that the algorithm for each node x of tree decomposition, calculate and stores the optimal solution of each valid sub-labeling of B_x on $G[V_x]$. Hence, in the last step of the algorithm, the optimal solution of each valid sub-labeling of B_r on $G = G[V_r]$ is achieved. Among these optimal solutions, one which does not contain the label *isolated* for all vertices in B_r and has maximum amount of F , is the final solution of MIN EXT WEC on (G, U) . Here we show that how F obtains the amount of optimal solution for each types of node and for every labeling in tree decomposition.

For the case that x is a **Leaf** node in the tree decomposition, it is trivial. Assume that x is an **Introduce** node with $B_x = B_y \cup \{a\}$ and suppose that the algorithm calculated and stored the optimal amount of F for all valid sub-labeling of B_y on $G[V_y]$. Concerning the labeling for vertex a (*center, leaf, isolated*) and all valid sub-labeling of B_y on $G[V_y]$ and also by considering U_x , we obtain optimal amount of F for all sub-labeling of B_x on $G[V_x]$: If $l(a) = \textit{isolated}$ and vertex

a is not an endpoint of U_x , then the amount of F is not changed and if $a \in V(U_x)$, since the final solution must contains U , the amount of F is ∞ ; If $l(a) = leaf$, the number of edges in U_x incident to a can not be more than 1 and for other cases we can easily calculate the amount of F ; If $l(a) = center$, for every neighbor v of a , $d_{U_x}(v) \leq 1$ and also a must have some neighbors M with label $leaf$ in B_x which have label $isolated$ in B_y . For all of these cases we calculate the amount of F and store the labeling.

Suppose node x is **Forget** and parent of node y in the tree decomposition such that $B_x = B_y \setminus \{a\}$. Since vertex a will not be appeared in the following steps of the algorithm, then we do not consider the cases that the label of vertex a was *isolated* among valid sub-labeling of B_y on $G[V_y]$, but by considering other cases (when the label of vertex a was *leaf*, *center*, we obtain an optimal amount of function F for all valid sub-labelings of B_x on $G[V_x]$.

Now assume that x is a **Join** node with two children y_1 and y_2 in the tree decomposition such that $B_x = B_{y_1} = B_{y_2}$. Suppose that the algorithm calculated and stored the optimal amount of F for all valid sub-labelings of B_{y_1} and B_{y_2} on $G[V_{y_1}]$ and $G[V_{y_2}]$ respectively. Recall that by construction of the tree decomposition, $V_{y_1} \cap V_{y_2} = B_x$ and also there is no edges between vertices of $V_{y_1} \setminus B_x$ and $V_{y_2} \setminus B_x$ in G . Hence, the optimal amount of F for all valid sub-labelings of B_x on $G[V_x]$ is obtained by checking all valid sub-labelings of B_{y_1} and B_{y_2} . A valid sub-labeling of B_x does not contain three following cases:

- 1– there is a vertex $v \in B_x$, with label *leaf* in B_{y_1} and B_{y_2} ,
- 2– there is a vertex $v \in B_x$, with label *leaf* in B_{y_1} and label *center* in B_{y_2} ,
- 3– there is a vertex $v \in B_x$, with label *leaf* in B_{y_2} and label *center* in B_{y_1} .

For all valid sub-labelings in B_x we calculate optimal amounts of F .

Assuming that the size of tree-width of the input graph is bounded by k . Since, for each vertex $v \in B_x$ there are three labels, we have 3^k different labelings for each B_x . Moreover, by considering the calculating method of F for each node type, the worst case for time complexity happens for **join nodes** where we have to consider three different cases when $l(u) = center$ and two different cases when $l(u) = leaf$ for each $u \in B_x$. Let z_1 (resp. z_2, z_3) be the number of vertices with labeling *center* (resp., *leaf*, *isolated*) in B_x for a join node x . All possible sub-labelings for B_x where $|B_x| = k$ can be computed as follows:

$$\sum_{z_1+z_2+z_3=k} \binom{k}{z_1, z_2, z_3} .3^{z_1} .2^{z_2} .1^{z_3} = (1 + 2 + 3)^k = 6^k \quad (3.3)$$

Thus, the time complexity of the proposed algorithm is $O^*(6^k)$. □

By changing function F of the algorithm such that putting *max* instead of *min* and $-\infty$ instead of ∞ , we can solve MAX EXT WEC.

Corollary 3.10. *MAX EXT WEC is solvable in polynomial time for bounded tree-width graphs.*

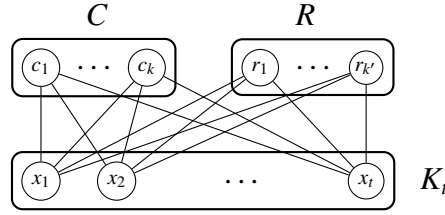


Figure 3.2: Illustration of the construction of the split graph $H = (X, (C \cup R), E_H)$.

If forced set U is a packing of non-trivial stars, then there exists some feasible solution for both MIN EXT WEC and MAX EXT WEC in complete graphs. Hence, in the next section, we will study both minimization and maximization versions of the problem on complete graphs.

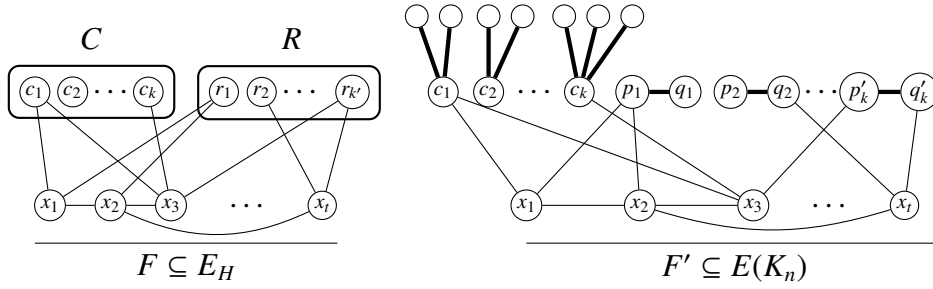
3.3 MIN EXT WEC in complete graphs

Let us start this section by a definition of a special transformation on weighted complete graphs, called *H-extended procedure*.

H-extended procedure In several parts of this section, we will consider the weighted graph $I = (H, w_H)$ built from an instance $I = (K_n, w, U)$ of the MIN EXT WEC where $U = S_U \cup M_U$ is a packing of non-trivial stars (see Figure 3.1). $H = (V_H, E_H)$ is a complete weighted split graph defined as follows:

- $V_H = X \cup (R \cup C)$ where $X = V(K_n) \setminus V(U)$, $C = \{c_1, \dots, c_k\}$ is the set of centers of stars in S_U and $R = \{r_1, \dots, r_{k'}\}$ is a set of vertices corresponding to stars in M_U .
- E_H is the set of edges of a complete split graph where the left side is a complete graph on X , the right side is an independent set on $R \cup C$ and we have a complete bipartite graph between them.
- $w_H(uv) = \begin{cases} w(uv) & \text{if } u, v \notin R \cup C, u \neq v \\ w(uv) & \text{if } u \notin R \cup C, v \in C \\ \min\{w(up_i), w(uq_i)\} & \text{if } u \notin R \cup C, v = r_i. (p_i, q_i, r_i \text{ are illustrated in Figure 3.1}) \end{cases}$

Figure 3.2 gives an illustration of the construction. The *H-extended procedure* transforms any subset $F \subseteq E(H)$ into a subset $F' \subseteq E(K_n)$ by adding the edges U . Precisely, each vertex $r_i \in R$ is replaced by a $K_{1,1}$ with endpoints p_i, q_i and each $c_i \in C$ is replaced by a star $K_{1,\ell}$, $\ell \geq 2$ with center c_i . Also, any edge $xr_i \in F$ is replaced by the edge xp_i if $w(xr_i) = w(xp_i)$, otherwise, by xq_i in F' . Obviously, these two constructions (H and H -extension procedure) are done in polynomial-time. Figure 3.3 proposes an example of the H -extended procedure. As one can see, instead of each vertex c_i of H , we add the corresponding star in the new graph. Moreover any r_i in H is replaced by an edge $p_i q_i$ and any edge $c_i x_j$ is replaced by either edge $p_i x_j$ or $q_i x_j$ not both.


 Figure 3.3: H -extended procedure. Bold edges are in U .

We now prove that the extended version of MIN EXT WEC can be much harder than the classical version, even in complete graphs. Actually, we will give a dichotomy result depending on parameter c of the c -relaxed (and extended c -relaxed) triangle inequality.

Theorem 3.11. *MIN EXT WEC in complete graphs is NP-hard for both extended c -relaxed (and c -relaxed) triangle inequality when $c > 1$.*

Proof. Let $c > 1$ be a constant. For both cases, we propose a reduction similar to what we did in Theorem 3.5. Given an instance $I = (\mathcal{X}, \mathcal{C})$ of SAT with variable set $\mathcal{X} = \{x_1, \dots, x_n\}$ and clause set $\mathcal{C} = \{c_1, \dots, c_m\}$, we build an instance $I' = (K_{2n+m}, w, U)$ of MIN EXT WEC as follows:

- $V(K_{2n+m}) = V(C) \cup V(X)$ where $V(C) = \{v_j : c_j \in \mathcal{C}\}$ and $V(X) = \{v_i^0, v_i^1 : i = 1, \dots, n\}$,
- $U = \{v_i^0 v_i^1 : x_i \in \mathcal{X}\}$,
- $M = \bigcup_{j=1}^m \{v_j v_i^1 : x_i \in c_j\} \cup \{v_j v_i^0 : \neg x_i \in c_j\}$.

Clearly, I' is built in polynomial-time. The weight function w , is defined $\forall xy \in E(K_{2n+m})$, by

$$w(xy) = \begin{cases} 0 & \text{if } xy \in U, \\ 1 & \text{if } xy \in M, \\ c & \text{if } xy \notin M, x \in V(C), y \in V(X), \\ 2c & \text{otherwise.} \end{cases}$$

We can easily verify that w satisfies the extended c -relaxed (and c -relaxed) triangle inequality. Let $\text{opt}_{\text{WEC}}(I')$ be the value of an optimal solution of MIN EXT WEC on the instance I' . We claim that I is satisfiable if and only if $\text{opt}_{\text{WEC}}(I') \leq m$.

Suppose I is satisfiable and let T be a satisfying truth assignment of I . For each clause c_j , we define $f(j)$ as an index of a variable $x_{f(j)}$ which satisfies clause c_j in T . We build a minimal edge cover S containing U such that $w(S) = \sum_{e \in S} w(e) = m$ as follows: $S = \{v_{f(j)}^1 v_j : c_j \in \mathcal{C} \wedge T(x_{f(j)}) = \text{true}\} \cup \{v_{f(j)}^0 v_j : c_j \in \mathcal{C} \wedge T(x_{f(j)}) = \text{false}\} \cup U$.

Conversely, assume that S is a minimal edge cover of K_{2n+m} containing U with $w(S) \leq m$. Since U is a matching of size n , and considering the weights of the graph, if S contains ℓ edges of weight $2c$, then $w(S) \geq 2c\ell + (m - 2\ell) = m + 2\ell(c - 1)$ because, these ℓ edges cover at most 2ℓ vertices of $V(C)$ and the weight of any edge incident to a vertex in $V(C)$ is at least 1 (recall $c > 1$). The term $(m - 2\ell)$ comes from the fact that each vertex in $V(C)$ must be covered, and each vertex in $V(C)$ must be incident to at most one edge of S by minimality, since $U \subseteq S$. Hence, we deduce $\ell = 0$. Now, if S contains ℓ' edges of weight $c > 1$, then these ℓ' edges cover exactly ℓ' vertices of $V(C)$ (since these edges connect a vertex of $V(C)$ to a vertex in $V(X)$) and $w(S) \geq \ell'c + (m - \ell') > m$. Evidently this comes from the fact that each vertex in $V(C)$ must be covered, and each vertex in $V(C)$ must be incident to at most one edge of S by minimality, since $U \subseteq S$. Hence, we deduce that $\ell' = 0$. Thus, S only contains unit weights. So, we define an initial truth assignment T by setting $T(x_i) = \text{true}$ for all $x_i \in \mathcal{X}$. For each $j(1 \leq j \leq m)$, by minimality v_j is incident to exactly one edge of S , since $S \subseteq U$ - let this edge be $v_j v'_i$. If $r = 0$, set $T(x_i) = \text{false}$. T is a well-defined truth assignment, since by minimality of S , and since $S \subseteq U$, it is impossible that $v_i^0 v_j \in S$ and $v_i^1 v_j \in S$ for any $x_i \in X$. Clearly T is a satisfying truth assignment for I . \square

Corollary 3.12. *The MIN EXT WEC for general weight function w is not approximable at all unless $P = NP$.*

Proof. Revisit the proof of Theorem 3.11 by replacing the weight of any edge e by $w'(e) = 0$ if $e \in (M \cup U)$ and $w'(e) = 1$ otherwise. Now, this restriction becomes the MIN EXT $\{0, 1\}$ -WEC and it is NP-complete to distinguish between $\text{opt}_{\text{WEC}}(I) = 0$ and $\text{opt}_{\text{WEC}}(I) > 0$ where $\text{opt}_{\text{WEC}}(I)$ is the size of optimal solution. \square

In the following, in order to get some positive results, we will use the MIN WEIGHTED LOWER-UPPER-COVER problem which is defined as follows:

MIN WEIGHTED LOWER-UPPER-COVER

Input: A weighted graph (V, E, w) and two non-negative integer functions a, b on V such that $\forall v \in V, 0 \leq a(v) \leq b(v) \leq d_G(v)$.

Solution: A subset $M \subseteq E$ such that the subgraph $G_M = (V, M)$ induced by M satisfies $a(v) \leq d_{G_M}(v) \leq b(v)$.

Output: Minimizing $w(M) = \sum_{e \in M} w(e)$. (if any)

The MIN WEIGHTED LOWER-UPPER-COVER problem is known to be solvable in polynomial-time (Theorem 35.2 Chapter 35 of Volume A in [151]). For an instance $I = (G, w)$ of the MIN WEIGHTED LOWER-UPPER-COVER problem, let $\text{opt}_{\text{LUC}}(I)$ denotes the value of an optimal solution.

Definition 3.13. *Let $I = (K_n, w, U)$ be an instance of MIN EXT WEC. Solution S is called claw U -free if for each claw $F = K_{1,3}$ subset of S , at least two edges of the claw F belongs to U , i.e. $|F \cap U| \geq 2$.*

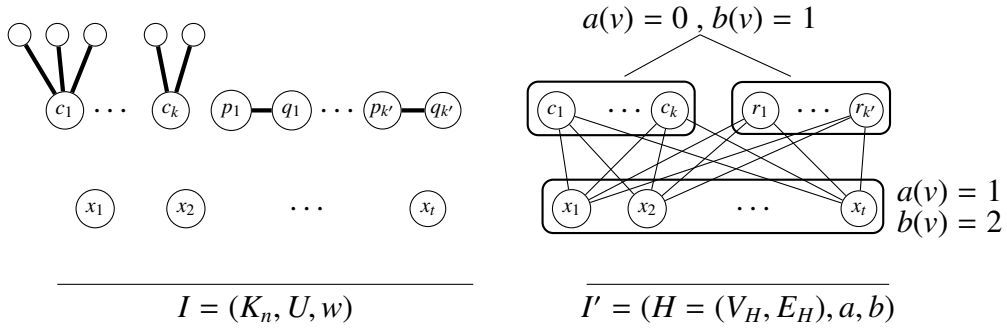


Figure 3.4: An instance I' of the MIN WEIGHTED LOWER-UPPER-COVER is shown on the right hand. Bold edges are in U .

Theorem 3.14. *MIN EXT WEC in complete graphs is solvable in polynomial-time for the c -relaxed and extended c -relaxed triangle inequality when $1/2 \leq c \leq 1$ and $c = 1$ respectively.*

Proof. We only deal with the c -relaxed triangle inequality case, because the other case is simpler. Let c be a constant with $1/2 \leq c \leq 1$. We solve MIN EXT WEC for the c -relaxed triangle inequality by solving MIN WEIGHTED LOWER-UPPER-COVER.

Let $I = (K_n, w, U)$ be an instance of the MIN EXT WEC where w satisfies the c -relaxed triangle inequality and let U be a packing of non-trivial stars. From I , we build an instance $I' = (H, w_H)$ where H is a complete split graph as described in *H-extended procedure*. Moreover, we consider two functions a, b of the MIN WEIGHTED LOWER-UPPER-COVER as follows: if $v \in V_H \setminus (R \cup C)$, then $a(v) = 1$ and $b(v) = 2$. Otherwise, $v \in R \cup C$ and $a(v) = 0$ and $b(v) = 1$. Figure 3.4 proposes an illustration of the construction.

By construction of I' , an optimal lower-upper-cover with parameters a, b is $\{P_4, C_3\}$ -free and then is an extended minimal edge cover of I . Hence, by letting opt_{LUC} and opt_{WEC} as the optimal value of MIN WEIGHTED LOWER-UPPER-COVER and MIN EXT WEC respectively, we have:

$$opt_{LUC}(I') \geq opt_{WEC}(I) \quad (3.4)$$

Conversely, let \mathcal{S}^* be an optimal solution of MIN EXT WEC of I . The next property allows us to focus on minimal edge cover claw U -free.

Property 3.15. *There is an optimal solution of MIN EXT WEC of I which is claw U -free.*

Proof. Let \mathcal{S} be an optimal solution of MIN EXT WEC which is not claw U -free and let $S = \{uv_i : i = 1, 2, 3\}$ be a claw in \mathcal{S} which is not U -free, w.l.o.g. suppose $uv_i \notin U$ for $i = 1, 2$. In particular, vertices v_1 and v_2 are not adjacent to U ; hence, $\mathcal{S}^* = (\mathcal{S} \setminus S) \cup \{v_1v_2, uv_3\}$ is a minimal edge cover with $w(\mathcal{S}^*) \leq w(\mathcal{S})$. By repeating this process for all such claws S in \mathcal{S} , we get the expected result. Note that if $c < 1$, all optimal solution of MIN EXT WEC are indeed claw U -free. \square

Hence, we can assume that \mathcal{S}^* is claw U -free, and then it is a lower-upper-cover with parameters a, b of I' :

$$\text{opt}_{\text{WEC}}(I) = w(\mathcal{S}^*) \geq w_H(\mathcal{S}^*) \geq \text{opt}_{\text{LUC}}(I') \quad (3.5)$$

Inequalities (3.4) and (3.5) give the expected result. \square

We finish this subsection by giving a parameterized complexity result depending on the number of forced edges.

Theorem 3.16. *MIN EXT WEC in complete graphs, parameterized by $|U|$ is in FPT and under the ETH, MIN EXT WEC cannot be solved in time $O^*(2^{o(|U|)})$*

Proof. Let $I = (K_n, w, U)$ be an instance of the MIN EXT WEC where we recall that $U = M_U \cup S_U$ with $M_U = \{p_i q_i : i = 1, \dots, k'\}$ and $S_U = \{F_1, \dots, F_k\}$. The set of centers is $C = \{c_1, \dots, c_k\}$ where c_i is the center of star F_i . As in Theorem 3.14, we solve several instances I_J of the MIN WEIGHTED LOWER-UPPER-COVER for each set $J \subseteq \{1, \dots, k'\}$. At the end, we return the solution minimizing $w(\mathcal{S}_J) = \text{opt}_{\text{LUC}}(I_J)$ among all possible sets J , that is $\mathcal{S} = \text{argmin}_J \text{opt}_{\text{LUC}}(I_J)$ where $\text{opt}_{\text{LUC}}(I_J)$ is the optimal value of the MIN WEIGHTED LOWER-UPPER-COVER problem on instance I_J .

Let $I = (K_n, w, U)$ be an instance of the MIN EXT WEC where U is a packing of stars. From I and a set $J \subseteq \{1, \dots, k'\}$, we built an instance $I_J = (H_J, w)$ where $H_J = (V_{H_J}, E_{H_J})$ is a complete subgraph of K_n and two functions a_J, b_J of the MIN WEIGHTED LOWER-UPPER-COVER as follows: $V_{H_J} = (V(K_n) \setminus V(U)) \cup (\{p_j : j \in J\} \cup \{q_j : j \in \{1, \dots, k'\} \setminus J\} \cup C)$. Finally, if $v \in V(K_n) \setminus V(U)$, then $a_J(v) = 1$ and $b_J(v) = d_{H_J}(v)$. Otherwise, for $v \in \{p_j : j \in J\} \cup \{q_j : j \in \{1, \dots, k'\} \setminus J\} \cup C$, $a_J(v) = 0$ and $b_J(v) = d_{H_J}(v)$. Let \mathcal{S}_J be an optimal solution of the MIN WEIGHTED LOWER-UPPER-COVER problem on (I_J, a_J, b_J) . Clearly, \mathcal{S}_J is a minimal edge cover on I and by construction there exists J^* such that $w(\mathcal{S}_{J^*}) = w(\mathcal{S}^*) = \text{opt}_{\text{WEC}}(I)$. The complexity of the whole algorithm is $O^*(2^{|M_U|}) = O^*(2^{|U|})$ and then MIN EXT WEC is in FPT.

The second part of the proof is a direct consequence of Corollary 3.12 for the MIN EXT $\{0, 1\}$ -WEC and ETH. Moreover, since the reduction is of linear size and by the Sparsification Lemma [111], we get the expected result. \square

Regarding Corollary 3.12 and Theorem 3.14, we focus on the approximation of the MIN EXT WEC for both c -relaxed (and extended c -relaxed) triangle inequality. Hence, let $c > 1$ be a fixed constant. We propose the following algorithm:

Algorithm 1: Approx 1

Input: $I = (K_n, w, U)$ where U is a packing of forced stars.

Output: A minimal edge cover \mathcal{S} of I containing U .

- 1 Build instance $I' = (H, w_H)$ from I as is described in *H-extended procedure*, where $H = (X, (R \cup C), E_H)$ is a split complete graph;
 - 2 Find an optimal solution $\mathcal{S}_H^* \subseteq E_H$ of the MIN WEIGHTED LOWER-UPPER-COVER problem on (I', a_1, b_1) with $a_1(v) = 1$ and $b_1(v) = d_H(v)$ if $v \in X$, $a_1(v) = 0$ and $b_1(v) = 1$ for $v \in R$ and $a_1(v) = 0$ and $b_1(v) = d_H(v)$ for $v \in C$;
 - 3 Convert \mathcal{S}_H^* into \mathcal{S} using the *H-extended procedure*;
 - 4 Return \mathcal{S} .
-

In Algorithm 1 we use optimal solution of the MIN WEIGHTED LOWER-UPPER-COVER problem as subroutine which is already explained in Theorem 3.14.

Theorem 3.17. *Algorithm 1 is a c -approximation of MIN EXT WEC in complete graphs for both c -relaxed (and extended c -relaxed) triangle inequality.*

Proof. Let $I = (K_n, w, U)$ be an instance of the MIN EXT WEC which satisfies the c -relaxed (resp. extended c -relaxed) triangle inequality. Clearly, for $c > 1$, a ρ -approximation algorithm for the case of extended c -relaxed triangle is also a ρ -approximation algorithm for the c -relaxed triangle. Hence, we will prove the correctness of Approx 1 only when the weight function satisfies extended c -relaxed triangle inequality. Considering the H -extended procedure which is already explained in the beginning of Section 3.3, we have:

- \mathcal{S} contains U and the degree of each vertex in \mathcal{S} is at least 1, since $a_1(v) = 1$ for $v \in X$.
- by the minimality and since $b_1(v) = 1$ for $v \in R$, \mathcal{S} is P_3 -free.

So, \mathcal{S} is a spanning star forest of K_n containing U .

Let \mathcal{S}^* be an optimal minimal weighted edge cover on I , we show how we can convert \mathcal{S}^* into a claw U -free $\mathcal{S}_H \subseteq E_H$ which is also a feasible solution of the MIN WEIGHTED LOWER-UPPER-COVER problem on (I', a_1, b_1) by losing at most a factor c .

For each $p_i q_i \in M_U$, let S_i^* is the star of \mathcal{S}^* containing edge $p_i q_i$; without loss of generality, assume that p_i is the center of S_i^* , and L_i^* denotes the leaves of S_i^* except q_i . If $|L_i^*| \geq 2$ and $x, y \in L_i^*$ are two distinct leaves, we replace edges $p_i x$ and $p_i y$ of S_i^* by two stars $S_i^* \setminus \{p_i x, p_i y\}$ and $\{xy\}$. By repeating this process until the star S_i^* (containing $p_i q_i$) satisfies $|L_i^*| \leq 1$ for each $p_i q_i \in M_U$, we obtain a packing of stars \mathcal{S}'_i such that $w(\mathcal{S}'_i) \leq cw(S_i^*)$ since $w(xy) \leq c(w(p_i x) + w(p_i y))$. By summing up these inequalities for all such stars, at the end we get a spanning star forest (minimal edge cover) \mathcal{S}_H (which is the union of \mathcal{S}'_i) with $w(\mathcal{S}_H) \leq cw(\mathcal{S}^*) = \text{opt}_{WEC}(I)$. This spanning star forest \mathcal{S}_H is by construction a feasible lower-upper-cover of (H, w_H) with parameters (a_1, b_1) . In conclusion,

$$w(\mathcal{S}) = w(\mathcal{S}_H^*) \leq w(\mathcal{S}_H) \leq cw(\mathcal{S}^*).$$

□

Corollary 3.18. *For complete graphs, MIN EXT WEC is in RAPX for both c -relaxed (and extended c -relaxed) triangle inequality.*

Proof. Using the proof of Theorem 3.17 and the extended c -relaxed triangle inequality, we deduce that Algorithm 1 is a residue c -approximation [161]. □

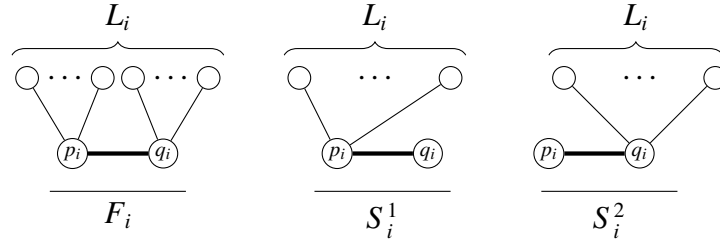


Figure 3.5: Illustration of construction of stars S_i^1 and S_i^2 from F_i . Bold edges are in \mathcal{U} .

We propose a new algorithm the following for the case that weight function w only satisfies the extended c -relaxed triangle inequality:

Algorithm 2: Approx 2

Input: $I = (K_n, w, U)$ where U is a packing of forced stars.

Output: A minimal edge cover \mathcal{S} of I containing U .

- 1 Build instance $I' = (H, w_H)$ from I as is described in *H-extended procedure*, where $H = (X, (R \cup C), E_H)$ is a split complete graph;
 - 2 Find an optimal solution $S_H^* \subseteq E_H$ of the MIN WEIGHTED LOWER-UPPER-COVER problem on (I', a_2, b_2) with $a_2(v) = 1$ and $b_2(v) = d_H(v)$ if $v \in X$ and $a_2(v) = 0$ and $b_2(v) = d_H(v)$ for $v \in R \cup C$;
 - 3 Convert S_H^* into S using the *H-extended procedure*;
 - 4 **for** (each connected component F_i of S with L_i as leaves such that $p_i q_i \in F_i$ and $L_i \cap \{p_i, q_i\} = \emptyset$) **do**
 - 5 Build two stars $S_i^1 = \{p_i x : x \in (L_i \cup \{q_i\})\}$ and $S_i^2 = \{q_i x : x \in (L_i \cup \{p_i\})\}$;
 - 6 **if** $w(S_i^1) \leq w(S_i^2)$ **then** $S \leftarrow (S \setminus F_i) \cup S_i^1$;
 - 7 **else** $S \leftarrow (S \setminus F_i) \cup S_i^2$;
 - 8 **Return** $S \leftarrow S$.
-

By construction, each connected component F_i of S with $p_i q_i \in F_i$ and $L_i \cap \{p_i, q_i\} = \emptyset$ has a diameter equals to 3 (some leaves are connected to p_i while the others leaves are connected to q_i). The other connected components are stars. Hence, \mathcal{S} is a minimal edge cover of I . Figure 3.5 proposes an illustration of the construction of stars S_i^1 and S_i^2 .

Theorem 3.19. *Algorithm 2 is a $\frac{c+1}{2}$ -approximation of MIN EXT WEC in complete graphs for extended c -relaxed triangle inequality.*

Proof. Let $I = (K_n, w, U)$ be an instance of the MIN EXT WEC which satisfies the extended c -relaxed triangle inequality. Assume that S contains $\ell \leq k'$ connected components F_1, \dots, F_ℓ during Step 4. of Algorithm 2 and without loss of generality, assume that $w(S_i^1) \leq w(S_i^2)$ for all $i \leq \ell$. Hence, the spanning star forest \mathcal{S} outputted by Algorithm 2 can be decomposed into S_1^1, \dots, S_ℓ^1 and \mathcal{S}' where $\mathcal{S}' = S \setminus (\cup_{i=1}^\ell F_i)$. By construction and using the extended c -relaxed triangle inequality, we have:

$$\begin{aligned}
 2w(S_i^1) &\leq w(S_i^1) + w(S_i^2) \leq \sum_{x \in L_i} w(p_i x) + \sum_{x \in L_i} w(q_i x) \leq \sum_{x \in L_i} w(p_i x) + w(q_i x) \\
 &\leq \sum_{x \in L_i} (c+1) \min\{w(p_i x), w(q_i x)\} \leq (1+c)w(F_i)
 \end{aligned} \tag{3.6}$$

Summing up inequality (3.6) for all $i \leq \ell$ and adding twice $w(S')$, we obtain:

$$\begin{aligned}
 2w(S) &= 2w(S') + 2 \sum_{i=1}^{\ell} w(S_i^1) \leq (1+c)w(S') + (1+c) \sum_{i=1}^{\ell} w(F_i) \\
 &\leq (1+c)w(S_H^*) = (1+c)opt_{LUC}(I').
 \end{aligned}$$

Let S^* be an optimal solution of MIN EXT WEC on I of value $opt_{WEC}(I)$; S^* is a feasible lower upper cover on (H, a_2, b_2) . Hence, we deduce $opt_{LUC}(I') \leq w(S^*) = opt_{WEC}(I)$. The two last inequalities provide the expected result. \square

Theorem 3.20. *For any $\epsilon > 0$ it is NP-hard to approximate and residue approximate the MIN EXT WEC in complete graphs within $\frac{7+c}{8} - \epsilon$ for both c -relaxed (and extended c -relaxed) triangle inequalities.*

Proof. Here, we use an in-approximability result of the optimization problem MAX 3-SAT. For an instance a 3-CNF-formula ϕ (i.e. with at most 3 variables in each clause), the goal is to find an assignment that satisfies the largest number of clauses. Let ϕ be an instance of MAX 3-SAT with variable set $X = \{x_1, \dots, x_n\}$ and clause set $C = \{c_1, \dots, c_m\}$. Without loss of generality, assume $m > n$ (otherwise, we duplicate the clauses c_1, \dots, c_m till $m > n$). Using the result of [101], we know, for every $\epsilon > 0$, given a 3-CNF-formula ϕ of 3-SAT, it is NP-hard to distinguish between two following cases:

- (Yes-instance) There exist an assignment satisfying $(1 - \epsilon)$ fraction of the clauses in ϕ .
- (No-instance) No assignment satisfies more than $(\frac{7}{8} + \epsilon)$ fraction of clauses in ϕ .

Let $\epsilon > 0$ and consider the reduction given in Theorem 3.11 for instances I_ϕ satisfying both c -relaxed (and extended c -relaxed) triangle inequalities when $c > 1$. Recall that

$$w(xy) = \begin{cases} 0 & \text{if } xy \in U, \\ 1 & \text{if } xy \in M, \\ c & \text{if } xy \notin M, x \in V(C), y \in V(X), \\ 2c & \text{otherwise.} \end{cases}$$

Completeness: Suppose there is an assignment of variables $\{x_1, \dots, x_n\}$ which satisfies $(1 - \epsilon)$ fraction of clauses (i.e. a Yes-instance). We build a minimal edge cover \mathcal{S} of n stars containing U as follows: suppose $T(\phi)$ is an assignment satisfies $(1 - \epsilon)m$ clauses = $C' = \{c_1, \dots, c_{(1-\epsilon)m}\}$. We start the construction of these n stars with total weight $(1 - \epsilon)m$ by taking the matching $\{v_i^0 v_i^1 : i = 1, \dots, n\}$ then, for each satisfied clause $c_j \in C'$ we connect v_j to either v_i^0 or v_i^1

depending on whether c_j is satisfied by $\neg v_i$ or v_i ; then, we complete these n stars into a minimal edge cover by connecting arbitrarily the ϵm remaining vertices (corresponding to other clauses which are not satisfied) to some centers. Hence, S is a minimal edge cover which contain U with $w(S) = (1 - \epsilon)m + c\epsilon m$, thus we have:

$$\text{opt}_{\text{WEC}}(I_\phi) \leq m(1 - \epsilon + c\epsilon) \quad (3.7)$$

Soundness: Consider an optimal minimal weighted edge cover S^* of I_ϕ with value $\text{opt}_{\text{WEC}}(I_\phi)$. Clearly, for each $1 \leq i \leq n$, at least one vertex of $\{v_i^0, v_i^1\}$ is a leaf of the star s_i^* of S^* . From S^* we build a truth assignment T for ϕ as follows: $T(x_i) = \text{true}$ if v_i^0 is a leaf and $T(x_i) = \text{false}$ otherwise. If vertex v_j is a leaf of star s_j^* , then clause c_j is satisfied by assignment T . Hence, the total weight of S^* is:

$$\text{opt}_{\text{WEC}}(I_\phi) \geq |\{c_j : c_j \text{ is satisfied}\}| + c|\{c_j : c_j \text{ is not satisfied}\}|$$

Furthermore, if at most $(\frac{7}{8} + \epsilon)m$ clauses are satisfied in ϕ , the weight of $\text{opt}_{\text{WEC}}(I_\phi)$ is at least:

$$\begin{aligned} \text{opt}_{\text{WEC}}(I_\phi) &\geq (\frac{7}{8} + \epsilon)m + (\frac{1}{8} - \epsilon)cm \\ &\geq m(\frac{7}{8} + \epsilon + \frac{1}{8}c - c\epsilon) \end{aligned} \quad (3.8)$$

The completeness and soundness arguments imply that it is NP-hard to distinguish whether instance I_ϕ has a minimal edge cover with weight at least $m(1 - \epsilon + c\epsilon)$ or at most $m(\frac{7}{8} + \epsilon + \frac{1}{8}c - c\epsilon)$. Therefore, we can conclude that it is NP-hard to approximate the MIN EXT WEC within factor:

$$\beta \geq \frac{\frac{7}{8} + \epsilon + \frac{1}{8}c - c\epsilon}{(1 - \epsilon + c\epsilon)} \quad (3.9)$$

By picking a small enough ϵ , we get the expected result. \square

3.4 MAX EXT WEC in complete graphs

Now, we study the maximization case when the weight function w is general, but non-negative and the graph is complete. Usually, the MAX STAR FOREST problem is defined in general graphs (i.e. not necessarily complete), and allowing trivial stars. This assumption is not restrictive because by completing the graph by weights 0, MAX EXT WEC becomes equivalent to MAX STAR FOREST problem. Moreover, by replacing the weights of required edges U by a large enough value, then MAX STAR FOREST problem and MAX EXT WEC are completely equivalent from a computational complexity point of view. However, these modifications affect the approximability of the problem. Hence, here we are interested in the hardest case which corresponds to $w(e) = 0$ for $\forall e \in U$. This means that the obtained results will be valid for the *residual approximation* [161]. Recall that $U = \{U_1, \dots, U_r\} = M_U \cup S_U$ where $r = k + k'$, $M_U = \{e_i : i = 1, \dots, k'\}$ is a matching of k' edges and $S_U = \{F_1, \dots, F_k\}$ is a set of k vertex-disjoint stars with at least two leaves each. The set of centers is $C = \{c_1, \dots, c_k\}$ and L_i are the leaves of F_i .

We study an intermediary problem called here EXTENDED DISJOINT SPANNING FOREST because it will provide an upper bound of our problem:

EXTENDED DISJOINT SPANNING FOREST

Input: A weighted connected graph (G, w) and a packing of non trivial stars $U = \{U_1, \dots, U_r\}$.

Solution: Spanning forest $S = \{S_1, \dots, S_r\} \subseteq E$ of G such that $U_i \subseteq S_i$.

Output: Maximizing $w(S) = \sum_{e \in S} w(e) = \sum_{i=1}^r \sum_{e \in S_i} w(e)$.

Obviously any feasible solution of MAX EXT WEC for an instance $I = (G, U)$ is a feasible solution for EXTENDED DISJOINT SPANNING FOREST, but the reverse is not essentially true because the solution is a collection of trees (not essentially stars). In the following we show that EXTENDED DISJOINT SPANNING FOREST is solvable in polynomial time.

Algorithm 3: Approx 3

Input: $I = (K_n, w, U)$ where U is a packing of forced stars.

Output: A minimal edge cover S of I containing U .

- 1 Build instance $I' = (G, w, U)$ of EXTENDED DISJOINT SPANNING FOREST;
 - 2 Find an optimal solution $S_1^* = \{S_1^*, \dots, S_r^*\}$ such that $U_i \subseteq S_i^*$ of EXTENDED DISJOINT SPANNING FOREST;
 - 3 **for** (each subtree S_i^*) **do**
 - 4 **if** $S_i^* = U_i$ **then** $S \leftarrow S \cup S_i^*$;
 - 5 **else**
 - 6 Split S_i^* into two spanning star forest (with possibly trivial stars) S_i^1 and S_i^2 such that $S_i^1 \cap S_i^2 = U_i$ and $S_i^1 \cup S_i^2 = S_i^*$ by dividing subtree S_i^* into alternating levels (even and odd from center of U_i);
 - 7 **if** $w(S_i^1) \geq w(S_i^2)$ **then** $S \leftarrow S \cup S_i^1$;
 - 8 **else** $S \leftarrow S \cup S_i^2$;
 - 9 Complete S into a minimal edge cover by connecting each isolated vertex to some center;
 - 10 Return $S \leftarrow S$.
-

Lemma 3.21. *There is a linear-time algorithm that solves EXTENDED DISJOINT SPANNING FOREST.*

Proof. The algorithm starts from U , sorts the edges by non increasing weights and iteratively adds edges satisfying condition $U_i \subseteq S_i$ for all $i \leq r$. The arguments of optimality are the standard ones and more generally are valid for matroids. We present them for sake of completeness. Consider a connected weighted graph $I = (G, w)$ and let S_1^* be an optimal solution of MAX EXT WEC, $S_1^* = \{S_1^*, \dots, S_r^*\}$ of G such that $U_i \subseteq S_i^*$. Consider $S_1 = \{S_1, \dots, S_r\}$ the solution returned by the greedy algorithm; let $e \in S_1^* \setminus S_1$. The addition of e^* to S_1 leads to either the creation of a cycle or the fusion of two subtrees S_ℓ and $S_{\ell'}$. In any case, all edges of that cycle or the path between U_ℓ and $U_{\ell'}$ are larger or equals than $w(e^*)$. \square

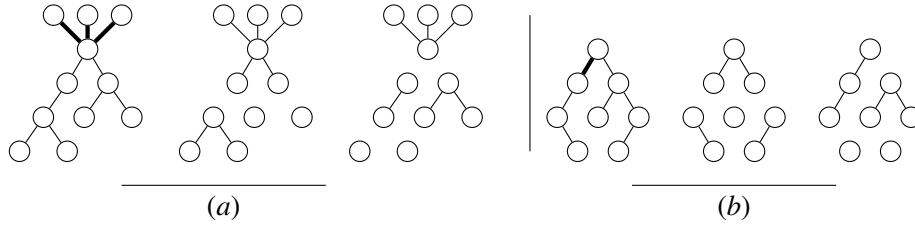


Figure 3.6: Construction of solutions S_i^1 and S_i^2 depending whether S_i^* contains $p_i q_i$ (case (b)) or not (case (a)); bold edges are in U_i . For each case, both solutions S_i^1 and S_i^2 are indicated (at this stage, trivial stars are allowed).

From $I = (K_n, w, U)$, we delete all edges $xy \notin U$ with $x \in L_i$ for some $i \leq k$ and $y \in V(K_n)$. Let $G = (V, E)$ be the resulting connected graph and $I' = (G, w, U)$ be the instance of EXTENDED DISJOINT SPANNING FOREST. Consider the Algorithm 3.

Let us formally explain how solutions are built during Step 5. Here, $U_i \subset S_i^*$; first we root subtree S_i^* at the center of U_i (if $U_i = \{p_i q_i\}$, we root S_i^* at p_i). Then, we construct a first partial solution which consider edges of $S_i^* \setminus U_i$ with odd levels and another partial solution with even levels. At the end of this Step 5. we add edges of U_i for both partial solutions. Figure 3.6 propose an illustration on the construction of the two spanning star forests (containing trivial stars at this stage) S_i^1 and S_i^2 of the induced subgraph (V, S_i^*) according to the structure of U_i .

Theorem 3.22. *Algorithm 3 is a $\frac{1}{2}$ -approximation of MAX EXT WEC in complete graphs.*

Proof. Let $I = (K_n, w, U)$ be an instance of the MAX EXT WEC, where $w(x, y) \geq 0$ for all $xy \in E(K_n)$. Consider a subtree $S_i^* \neq U_i$ of the optimal solution \mathcal{S}_1^* found by Algorithm 3 for the EXTENDED DISJOINT SPANNING FOREST problem on instance (G, w) . Clearly, either $S_i^* \setminus U_i$ has two subtrees (one containing p_i , the other containing q_i) and in this case $U_i = \{p_i q_i\}$ or $S_i^* \setminus U_i$ remains a subtree. Without loss of generality, assume that $w(S_i^1) \geq w(S_i^2)$ (Step 7 of Algorithm 3); since S_i^1 and S_i^2 are two spanning star forests (possibly containing trivial stars at this stage) such that $S_i^1 \cap S_i^2 = U_i$ and $S_i^1 \cup S_i^2 = S_i^*$, we get: $\forall i \leq r, 2w(S_i^1) \geq w(S_i^1) + w(S_i^2) \geq w(S_i^*) + w(U_i)$. By summing up inequality (3.10) for all $i \leq \ell$ (note this inequality also holds when $S_i^* = U_i$), and using $w(e) \geq 0$, we obtain:

$$2w(\mathcal{S}) \geq \sum_{\ell=1}^r w(\mathcal{S}_i^*) = w(\mathcal{S}_1^*) \quad (3.10)$$

On the other hand, suppose $\mathcal{S}^* = \{s_1^*, s_2^*, \dots, s_m^*\}$ is an optimal solution of MAX EXT WEC on I with value $w(\mathcal{S}^*) = \text{opt}_{\text{WEC}}(I)$, and suppose only r starts $r \leq m$ among \mathcal{S}^* have some intersections with U . In this case, we make a particular feasible solution of EXTENDED DISJOINT SPANNING FOREST on $I' = (G, w, U)$ by adding $m - r$ new edges which connect the $m - r$ remaining stars to the center of star which has an intersection with U . Since the weights are non-negative, we have:

$$w(\mathcal{S}_1^*) \geq \text{opt}_{\text{WEC}}(I) \quad (3.11)$$

combining inequalities (3.10) and (3.11) gives the result. \square

Setting $w(e) = 0$ for $e \in U$ leads to the following corollary.

Corollary 3.23. *MAX EXT WEC in complete graphs is in RAPX.*

Theorem 3.24. *For any $\epsilon > 0$ it is NP-hard to approximate and residue approximate MAX EXT WEC in complete graphs within $\frac{7}{8} + \epsilon$.*

Proof. The reduction given in Corollary 3.12 is indeed a reduction preserving approximation from 3-SAT to MAX {0, 1}-EXT WEC. Hence, using [101], we obtain the result. \square

So far, we have studied both maximization and minimization versions of extended edge cover problems in general and complete graphs. In the following, we will study MAX EXT WEC in the particular case $U = \emptyset$ for some different graph classes.

3.5 Weighted upper edge cover

In this section we study a particular case of MAX EXT WEC, when the forced set is an empty-set. Formally, the associated optimization problem is called WEIGHTED UPPER EC and asks to find the largest weighted minimal edge cover of an edge-weighted graph.

WEIGHTED UPPER EC

Input: A weighted connected graph $G = (V, E, w)$, where $w(e) \geq 0$ for $e \in E$.

Solution: Minimal edge cover $S \subseteq E$.

Output: Maximize $w(S) = \sum_{e \in S} w(e)$.

The unweighted variant of the problem denoted by UPPER EC is equivalent to MIN DS in general graphs without isolated vertices [137]. Let $opt_{UEC}(G)$ and $\gamma(G)$ be the sizes of optimal solution of UPPER EC and MIN DS for a given graph $G = (V, E)$ respectively, then $opt_{UEC}(G) = |V| - \gamma(G)$. We will consider the implications of this important remark afterwards in the section.

In the beginning of this chapter, we defined the optimization problem MAX STAR FOREST problem, which consists in a weighted graph $G = (V, E, w)$ as an input, and the goal is to find a spanning star forest of maximum total weight. The un-weighted version (corresponding to the case $w(e) = 1$ for all edges e) of MAX STAR FOREST problem, denoted by MAX {1}-STAR FOREST PROBLEM, seeks a spanning star forest with maximum number of leaves. Let $opt_{\{1\}-SF}(G)$ denotes the value of MAX {1}-STAR FOREST PROBLEM for a given graph G then, $opt_{\{1\}-SF}(G) = opt_{UEC}(G)$, since any spanning star forest (with possible trivial stars) can be (polynomially) converted into a star spanning forest without trivial stars (i.e. a minimal edge cover) with same size [137]. Hence, these two problems are completely equivalent even from an approximation point of view.

Concerning edge-weighted graphs, the relationship between WEIGHTED UPPER EC and MAX STAR FOREST Problem is less obvious. Let $opt_{SF}(G)$ and $opt_{UWEC}(G)$ be the optimal values of MAX STAR FOREST problem and WEIGHTED UPPER EC of a given weighted graph $G = (V, E, w)$ respectively, we only have $opt_{SF}(G, w) \geq opt_{UWEC}$, because any minimal edge cover is a particular

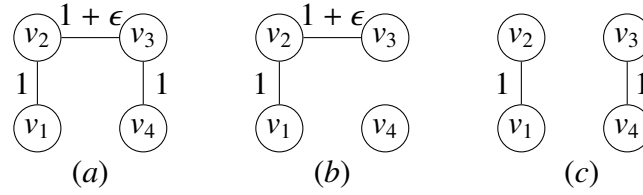


Figure 3.7: (a) : The weighted graph $G = (V, E, w)$. (b) : The optimal solution of MAX STAR FOREST problem with value $2 + \epsilon$. (c) : The optimal solution of WEIGHTED UPPER EC with value $\text{uec}(G, w) = 2$.

spanning star forest. However, the difference between these two values can be arbitrarily large as indicated in Figure 3.7 (in the graph drawn in Figure 3.7.(b), v_4 is an isolated vertex) when ϵ goes to Infinity. This means that isolated vertices play an important role in feasible solutions.

Given a spanning star forest $S = \{S_1, \dots, S_r\}$ of $(G = (V, E, w))$, we rename vertices such that there is some p , $0 \leq p < r$ such that $S_i = \{v_i\}$ are trivial stars for all $1 \leq i \leq p$ (if $p = 0$, then there is no trivial stars), and S_j are non-trivial stars whose c_j is the center for all $j > p$ (if S_j is a single edge, both endpoints are considered as possible centers). We define $\text{Triv} = \{v_i : i \leq p\}$ as the set of isolated vertices of $(V, E(S))$ where $E(S) = \cup_{j>p}^r S_j$; moreover, V_l and V_c are respectively the set of leaves and the set of centers of stars in $V \setminus \text{Triv}$. Finally, for $v \in V_l$, $e_v(S) = c'v \in E(S)$ denotes the edge linking the center c' to the leaf v .

We mainly focus on specific solutions of MAX STAR FOREST problem called *nice spanning star forests* defined as follows:

Definition 3.25. S is a nice spanning star forest of $G = (V, E, w)$ if $\text{Triv} = \{v_i : i \leq p\}$ is an independent set in G and all edges of G starting at Triv are linked to leaves of some ℓ -stars of S with $\ell \geq 2$. Moreover, $w(uv) \leq w(e_v(S))$ for $u \in \text{Triv}$, $v \in V_l$.

Property 3.26. Any spanning star forest of $G = (V, E, w)$ can be polynomially converted into a nice one with at least the same weight.

Proof. The weights of (G, w) are non-negative. Thus, if Triv is not an an independent set or if some vertex of Triv is linked to some center of S , we could obtain a better spanning star forest with less isolated vertices. In particular, it implies that no vertex of Triv is linked to a 1-star (i.e. a K_2 of S). Finally, if $w(uv) > w(e_v(S))$, then $S' = (S \setminus \{e_v(S)\}) \cup \{uv\}$ is a better spanning star forest. \square

In the remaining part of this chapter, we will study the approximability of WEIGHTED UPPER EC in some different graph classes.

3.5.1 Weighted Upper EC in complete graphs

In this section, we deal with edge-weighted complete graphs. This case seems to be the simplest one, because the equivalence between WEIGHTED UPPER EC and MAX STAR FOREST problem

for the unweighted case proven in [137] remains valid for the weighted case as proven in the following.

Theorem 3.27. *MAX STAR FOREST problem in general graphs is equivalent to approximate WEIGHTED UPPER EC in complete graphs.*

Proof. We propose two approximation preserving reductions, one from MAX STAR FOREST problem in general graphs to WEIGHTED UPPER EC in complete graphs and the other from WEIGHTED UPPER EC to MAX STAR FOREST problem in complete graphs.

- Reduction from MAX STAR FOREST problem to WEIGHTED UPPER EC in complete graphs.

Let (G, w) be an instance of MAX STAR FOREST problem where $G = (V, E)$ is a connected graph with n vertices, edge-weighted using w . We build an instance (K_n, w') of WEIGHTED UPPER EC where K_n is an edge-weighted complete graph $(V, E(K_n))$ over n vertices, edge-weighted with w' , by keeping same weight for existing edges of G and assigning weight 0 for any non-existing edge of G . In other words, $\forall u, v \in V$ with $u \neq v$, $w'(uv) = w(uv)$ if $uv \in E$ and $w'(uv) = 0$ otherwise.

Let opt_{SF} and opt_{WUEC} be values of optimal solution of MAX STAR FOREST problem and WEIGHTED UPPER EC problems respectively. Suppose that $S' \subseteq E(K_n)$ is a minimal edge cover of WEIGHTED UPPER EC with weight $w'(S')$. The restriction of S' to G gives a star spanning forest (with eventually trivial stars) S . Obviously, by construction we have:

$$w(S) = w'(S') \quad (3.12)$$

Thus, from equality (3.12), we deduce $opt_{SF}(G, w) \geq opt_{WUEC}(K_n, w')$. Conversely, assume that S^* be an optimal star spanning forest of MAX STAR FOREST problem on (G, w) . By adding some edges from the center of some stars to the isolated vertices of S^* , we obtain a minimal edge cover of K_n of at least same value. Hence, $opt_{WUEC}(K_n, w') \geq opt_{SF}(G, w)$. So, we can deduce:

$$opt_{WUEC}(K_n, w') = opt_{SF}(G, w) \quad (3.13)$$

Using equations (3.12) and (3.13), we can deduce that any ρ approximation of WEIGHTED UPPER EC for (K_n, w') can be polynomially converted into a ρ approximation of MAX STAR FOREST problem for (G, w) .

- Reduction from WEIGHTED UPPER EC to MAX STAR FOREST problem in complete graphs.

From an instance (K_n, w) of WEIGHTED UPPER EC, we set $(G, w') = (K_n, w)$ as an instance of MAX STAR FOREST problem. Since the graph is complete, the weights are non-negative and the goal is maximization, we can only consider star spanning forests without trivial stars, i.e. minimal edge covers. Hence, WEIGHTED UPPER EC can be seen as a sub-problem of MAX STAR FOREST problem, even from an approximation point of view. \square

From Theorem 3.27 and using known results for MAX STAR FOREST problem given in [143, 44], we have:

Corollary 3.28. *In complete graphs, WEIGHTED UPPER EC is $1/2$ -approximable but not approximable within $\frac{10}{11} + \varepsilon$ unless $P = NP$.*

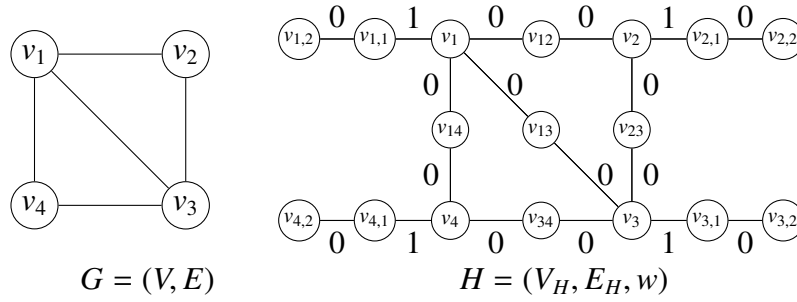


Figure 3.8: Construction of H from G . The binary weights are indicated on edges.

3.5.2 Weighted Upper EC in bipartite graphs

Let us now focus on bipartite graphs. We prove that, even in bipartite graphs with binary weights, WEIGHTED UPPER EC is not $O(n^{\frac{1}{2}-\epsilon})$ approximable unless $P = NP$. Also, we show the problem is APX-complete even for bipartite graphs with fixed maximum degree Δ .

Theorem 3.29. *WEIGHTED UPPER EC in bipartite graphs with binary weights and the cycle inequality is as hard² as MAX IS in general graphs.*

Proof. We propose an approximation preserving APX-reduction from MAX IS to WEIGHTED UPPER EC. Computing $\alpha(G)$ (the size of the largest set of independent vertices in G) is NP-hard and it is not constant approximable within polynomial-time [88].

Given a connected graph $G = (V, E)$ with n vertices and m edges where $V = \{v_1, \dots, v_n\}$, instance of MAX IS, we build a connected bipartite edge-weighted graph $H = (V_H, E_H, w)$ as follows (see also Figure 3.8):

- For each $v_i \in V$, add a P_3 with edge set $\{v_i v_{i,1}, v_{i,1} v_{i,2}\}$.
 - For each edge $e = v_i v_j \in E$ where $i < j$, add a middle vertex v_{ij} on edge e .
- $$w(e) := \begin{cases} 1 & \text{if } e = v_i v_{i,1} \text{ for some } v_i \in V \\ 0 & \text{otherwise.} \end{cases}$$

Clearly, H is a connected bipartite graph on $|V_H| = 3n + m$ vertices and $|E_H| = 2(m + n)$ edges. Moreover, the weight function is binary and the instance satisfies the cycle inequality.

Let S^* be a maximum independent set of G with size $\alpha(G)$. For each $e \in E$, let $v^e \in V \setminus S^*$ be a vertex which covers e ; it is possible since $V \setminus S^*$ is a *vertex cover* of G . Moreover, $\{v^e : e \in E\} = V \setminus S^*$ since S^* is a maximum independent set of G . Now, $S' = \{v^e v^e : e \in E\} \cup \{v_i v_{i,1} : v_i \in S^*\}$ is a minimal edge cover of H . By construction, $w(S') = |S^*| = \alpha(G)$. Hence, we deduce:

²The reduction is actually a Strict-reduction and it is a particular A-reduction which preserves constant approximation.

$$\text{opt}_{WUEC}(H, w) \geq \alpha(G) \quad (3.14)$$

Conversely, suppose S' is a minimal edge cover of H with weight $w(S')$. Let us make some simple observations of every minimal edge cover of H . Clearly, $\{v_i v_{i2} : v_i \in V\}$ is part of every feasible solution because v_{i2} for $v_i \in V$ are leaves of H . Moreover, for each $e = v_i v_j \in E$ with $i < j$, at least one edge between $v_i v_{ij}$ or $v_j v_{ij}$ belongs to any minimal edge cover of H . If $v_i v_{ij} \notin S'$, it implies that $v_j v_{j,1} \notin S'$ is not a part of the feasible solution because of minimality of S' . Hence, $S = \{v_i : v_i v_{ij} \in S'\}$ is an independent set of G with size $|S| = w(S')$. We deduce,

$$\alpha(G) \geq \text{opt}_{WUEC}(H, w) \quad (3.15)$$

Using inequalities (3.14) and (3.15) we deduce:

$$\alpha(G) = \text{opt}_{WUEC}(H, w) \quad (3.16)$$

In conclusion, for each minimal edge cover S' on H , there is an independent set S of G (computed in polynomial-time) such that $|S| \geq w(S')$. \square

From Theorem 3.29, we immediately deduce that WEIGHTED UPPER EC in bipartite graphs is not in APX unless $P = NP$. However, using several results [88, 4] concerning the APX-completeness of MAX IS in connected graph G with constant maximum degree $\Delta(G) \geq 3$ or NP-completeness of MAX IS in planar graphs, we obtain:

Corollary 3.30. *WEIGHTED UPPER EC in bipartite (resp. planar bipartite) graphs of maximum degree Δ for any fixed $\Delta \geq 4$ and binary weights is APX-complete (resp. NP-complete).*

Proof. Let us revisit the construction given in Theorem 3.29. If the instance of MAX IS has maximum degree 3 (resp. is planar with maximum degree 3), then the constructed instance of WEIGHTED UPPER EC is a bipartite (resp. planar bipartite) graph of maximum degree 4. \square

Using the strong in-approximability result for MAX IS given in [165], and by the reduction given in the previous theorem which is a gap-reduction, we also deduce:

Corollary 3.31. *For any $\varepsilon > 0$, WEIGHTED UPPER EC in bipartite graphs of n vertices is not $O(n^{\varepsilon - \frac{1}{2}})$ approximable unless $P = NP$, even for binary weights and cycle inequality.*

Proof. We use the reduction given in Theorem 3.29 and the inapproximability of MAX IS. MAX IS is known to be, in a sense, completely non-approximable [165]. In particular, it is known that, for all $\varepsilon > 0$, it is NP-hard to distinguish for an n -vertex graph G between $\alpha(G) > n^{1-\varepsilon}$ and $\alpha(G) < n^\varepsilon$.

In the construction of H (see Figure 3.8), we know that $|V_H| = m + 3n$ and $|E_H| = 2(m + n)$ where m, n are numbers of the edges and vertices of G respectively. Hence, we deduce $|V_H| \leq 2n^2$, and the claimed result follows. \square

We can also deduce one inapproximability result depending on the maximum degree.

Corollary 3.32. *For any constant $\varepsilon > 0$, unless $NP \subseteq ZPTIME(n^{\text{poly log } n})$, it is hard to approximate WEIGHTED UPPER EC on bipartite graphs of maximum degree Δ within a factor of $\Theta\left(\frac{1}{\Delta^{1-\varepsilon}}\right)$.*

Proof. We will prove that it is difficult for a graph H (even bipartite with binary weights) of maximum degree Δ to distinguish between the following two cases:

- (Yes-Instance) $\text{uec}(H, w) \geq \frac{|V(H)|}{\Delta(G)^{1+\varepsilon}}$,
- (No-Instance) $\text{uec}(H, w) \leq \frac{|V(H)|}{\Delta(G)^{2-\varepsilon}}$.

Hence, the result consists of showing that the transformation given in Theorem 3.29 is a gap reduction. In [45], it is proved that let $\tau(n)$ be any function from integers to integers and assuming $NP \not\subseteq ZPTIME(n^{O(\tau(n))})$, then there is no polynomial-time algorithm that can solve the following problem (Theorem 5.7, adapted from [154]).

For any constant $\varepsilon > 0$ and any integer q , given a regular graph G of size $q^{O(\tau(n))}$ such that all vertices have degree $\Delta = 2^{O(\tau(n))}$, the goal is to distinguish between the following two cases:

- (Yes-Instance) $\alpha(G) \geq \frac{|V(G)|}{\Delta^\varepsilon}$,
- (No-Instance) $\alpha(G) \leq \frac{|V(G)|}{\Delta^{1-\varepsilon}}$.

Note that if G is a Δ -regular graph, then graph H resulting of Theorem 3.29 is a bipartite graph of maximum degree $\Delta + 1 = \Theta(\Delta)$. Thus, since $\alpha(G) = \text{uec}(H, w)$ and $|V(H)| = 3|V(G)| + |E(G)| = \Theta(\Delta|V(G)|)$, we get the expected result. \square

Let us note that in the case of regular bipartite graphs, we can easily get a simple positive approximation result. A graph is called k -regular if all vertices have the same degree k . For such bipartite graph $G = (V, E)$, it is well known that G is k -factorable that is E is decomposable into k perfect matchings (E_1, \dots, E_k) . Since, any perfect matching is a particular minimal edge cover, we deduce the maximum weighted perfect matching M of the graph satisfies $k \times w(M) \geq \sum_{i=1}^k w(E_i) = w(E) \geq \text{uec}(G, w)$. In conclusion, there is a polynomial $\frac{1}{\Delta}$ -approximation for WEIGHTED UPPER EC in Δ -regular bipartite graphs. A $\frac{1}{2\Delta}$ -approximation for general graphs of bounded degree Δ will be given later (Section 3.5.5 Corollary 3.44).

3.5.3 Weighted Upper EC in split graphs

We will now focus on split graphs. Recall that a graph $G = (L \cup R, E)$ is a split graph if the subgraph induced by L and R is a maximum clique and an independent set respectively. A graph is called Δ -subregular split graph if for $v \in L$, $d_G(v) \leq \Delta + |L| - 1$ and for $v \in R$, $d_G(v) \leq \Delta$. This means that the graph induced by crossing edges is of maximum degree at most Δ . We prove that WEIGHTED UPPER EC in split graphs as well as bipartite graphs which has already been proved above, is as hard as MAX IS in general graphs in terms of approximation.

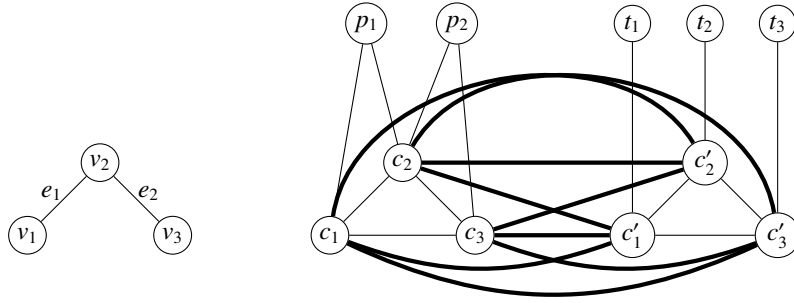


Figure 3.9: Construction of split graph $H = (V_H, E_H)$ from a P_3 . The weights of thick edges in H are 1 and for the others are 0.

Theorem 3.33. *WEIGHTED UPPER EC in split graphs with binary weights and cycle inequality is as hard³ as MAX IS in general graphs.*

Proof. The proof is based on a reduction from MAX IS. Given a graph $G = (V, E)$ of n vertices and m edges where $V = \{v_1, \dots, v_n\}$ and $E = \{e_1, \dots, e_m\}$, instance of MAX IS, we build a split weighted graph $H = (V_H, E_H, w)$ as follows:

- Put two copies of vertices V in H , indicated by $C = \{c_1, \dots, c_n\}$ and $C' = \{c'_1, \dots, c'_n\}$ and make two cliques of size n such that all pairs of vertices in C and C' are connected to each other with edges of weight 0.
- Connect all pairs $c_i c'_j$ for $1 \leq i, j \leq n$ with edges of weight 1 to make a clique of size $2n$.
- Add a set of m new vertices $\{p_1, \dots, p_m\}$ corresponding to edges of E and connect p_i to c_j, c_k with edges of weight 0 if $e_i = v_j v_k \in E$.
- Add a set of n new vertices $\{t_1, \dots, t_n\}$ and connect each t_i to c'_i with edges of weight 0.

It is easy to check that H is a weighted split graph with binary weights and cycle inequality which contains a clique of size $2n$ and an independent set of size $n + m$. Figure 3.9 gives an illustration of the construction of H from a P_3 . Now, we claim that G has an independent set of size k if and only if there exists a minimal edge cover of H with total weight k .

Let S be an independent set of G with size $|S|$. For each $e_i \in E$, there is $v_{e_i} \notin S$ which covers e_i , since S is an independent set of G . Consider the set $\{c_{e_i} : v_{e_i} \notin S\}$ of vertices in C corresponding to vertices of $V \setminus S$. $S' = \{c_{e_i} p_i : e_i \in E\} \cup \{c'_i t_i : v_i \in V\} \cup \{c_i c'_i : v_i \in S^*\}$ is a minimal edge cover of H . By construction, $w(S') = |S|$. Let opt_{WUEC} and α denotes the values of optimal solution of WEIGHTED UPPER EC and MAX IS respectively then, we deduce:

$$opt_{WUEC}(H, w) \geq \alpha(G) \quad (3.17)$$

³The reduction is actually a Strict-reduction and it is a particular A-reduction which preserves constant approximation.

Conversely, let be a minimal edge cover of H with weight $w(S')$. Since, for $1 \leq i \leq n$, t_i s are leaves in H , $\{t_i c'_i : v_i \in V\}$ is a part of S' . Moreover, for each $e_k = v_i v_j \in E$ with $i < j$, at least one edge among $c_i p_k$ or $c_j p_k$ belongs to S' . Without loss of generality, assume that $c_i p_k \in S'$; this means that $c_i c'_j \notin S'$ for all $1 \leq j \leq n$. Furthermore, for each $c_i \in C$ at most one edge $c_i c'_j \in S'$ for $1 \leq j \leq n$. Hence, $S = \{v_i : c_i c'_j \in S'\}$ is an independent set of G with size $|S| = w(S')$. We deduce,

$$\alpha(G) \geq \text{opt}_{WUEC}(H, w) \quad (3.18)$$

Using inequalities (3.17) and (3.18) we deduce $\alpha(G) = \text{opt}_{WUEC}(H, w)$. \square

If G has a maximum degree 3, then (H', w) is a 3-subregular split graph. Similar results to Corollary 3.30 and Corollary 3.31 can be derived:

Corollary 3.34. *WEIGHTED UPPER EC in 3-subregular split graphs is APX-complete and for any $\varepsilon > 0$, WEIGHTED UPPER EC in split graphs of n vertices is not $O(n^{\varepsilon - \frac{1}{2}})$ approximable unless $P = NP$.*

3.5.4 Weighted Upper EC in k -trees

Recall that a k -tree is a graph which results from the following inductive definition:

- A K_{k+1} is a k -tree,
- If a graph G is a k -tree, then the addition of a new vertex which has exactly k neighbors in G such that these $k + 1$ vertices induce a K_{k+1} forms a k -tree.

As a main result in this section, we prove WEIGHTED UPPER EC is APX-complete in weighted-dense k -trees even for binary weights.

Negative approximation result

From Corollary 3.28, we already know that WEIGHTED UPPER EC is NP-hard to approximate within a ratio strictly better than $\frac{10}{11}$ because the class of all k -trees contains the class of complete graphs. However, this lower bound needs a non-constant number of distinct values [44]. Moreover, in Theorem 3.27, we showed that WEIGHTED UPPER EC in weighted complete graphs is equivalent to MAX STAR FOREST problem in general graphs. In [143, Theorem 3.6], it is proved that MAX STAR FOREST problem is hard to approximate in general graphs within ratio $\frac{259}{260} + \varepsilon$ for any $\varepsilon > 0$, so WEIGHTED UPPER EC in complete graphs and k -trees with binary weights is not strictly approximable within ratio better than $\frac{259}{260} \approx 0.9961$. Here, we propose a new approximation preserving reduction for WEIGHTED UPPER EC in k -trees. Our reduction does not improve the existed bound $\frac{259}{260}$, but help us to find some new upper bounds for WEIGHTED UPPER EC in weighted-dense k -trees and UPPER EC in dense graphs.

Recall that a graph $G = (V, E)$ with $|V| = n$ is called c -dense if $|E| \geq \frac{cn^2}{2}$ [10, 104]. This concept can be adapted to edge-weighted maximization problems as follows. For a non-negative

edge-weighted graph (G, w) , we assume $w(x, y) = 0$ for a non-edge $xy \notin E$ and for all the p distinct weights $w_i > 0, i = 1, \dots, p$ of the instance, we denote by \bar{w} the *average weight* such that $\bar{w} = \frac{\sum_{i=1}^p w_i}{p}$.

Definition 3.35. An edge weighted graph $G = (V, E, w)$ with $w \geq 0$ is c -weighted-dense if

$$\sum_{x, y \in V} w(x, y) \geq cn^2 \times \bar{w}$$

This notion captures the fact that a huge number of edges with average weight are available in the graph. In particular, using Definition 3.35, we deduce that a c -dense graph is also weighted c -dense by taking $w(xy) = 1$ if $xy \in E$.

Theorem 3.36. WEIGHTED UPPER EC is APX-hard in the class of weighted c -dense k -trees, even for binary weights and $c = \frac{4}{25}$.

Proof. We give an approximation preserving reduction from independent set problem. It is known that MAX IS is APX-complete in graphs of maximum degree Δ with $\Delta \geq 3$ [4].

Let $G = (V, E)$ be an instance of MAX IS where G is a connected graph of maximum degree Δ of $n \geq 3$ vertices and m edges. We build a weighted dense graph $G' = (V', E', w)$ for WEIGHTED UPPER EC problem where $V' = V_C \cup V'_E$ and $E' = E'_C \cup (\cup_{e \in E} E'_e)$ as follows:

- $V'_C = \{v' : v \in V\}$ and $V'_E = \cup_{e \in E} V'_e$ where $V'_e = \{v_{e_1}, \dots, v_{e_x}\}$ for any integer $1 \leq x \leq n - 1$.
- The subgraph $G'[V'_C] = (V'_C, E'_C)$ induced by V'_C is a K_n .
- For each $e = uv \in E$, let us describe the edge set E'_e :
 - for every $i = 1, \dots, x$, vertex v_{e_i} is linked to u' and v' .
 - vertex v_{e_1} is linked to the subset $S_{e_1} = V'_C \setminus \{u', v'\}$.
 - for every $i = 2, \dots, x$, vertex v_{e_i} is linked to $\{v_{e_1}, \dots, v_{e_{i-1}}\}$ and an arbitrary subset $S_{e_i} \subset S_{e_{i-1}}$ of size $n - i - 1$.

The weight $w(yz)$ for $yz \in E'$ is assigned by:

$$w(yz) = \begin{cases} x & yz \in E'_C, \\ 1 & yz \in E'_e \text{ with } e = uv \in E \text{ and } y \in \{u', v'\}, z \in V'_e, \\ 0 & \text{otherwise.} \end{cases}$$

Note that $|V'| = n + mx$ and clearly G' can be constructed from G in polynomial time. G' is a n -tree because initially all $V'_C \cup \{v_{e_1}\}$ are clique of size $n + 1$ for $e \in E$ and at each step the addition of $v_{e, i+1}$ maintains a K_{n+1} containing $v_{e, i+1}$ in the subgraph induced by $V'_C \cup \{v_{e_j} : e \in E, j \leq i\}$. Figure 3.10 proposes an illustration of this construction for a P_3 where $x = 2$.

We are going to prove that any ρ -approximation for WEIGHTED UPPER EC in k -trees can be polynomially converted into a $(1 + \frac{\Delta^2}{2})\rho - \frac{\Delta^2}{2}$ approximation ratio for MAX IS in graphs of maximum degree Δ .

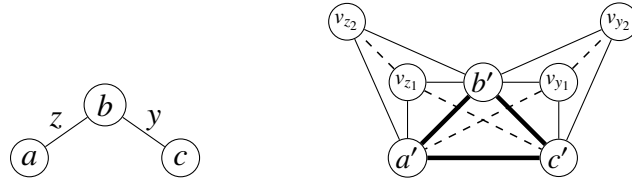


Figure 3.10: The constructed weighted graph $G' = (V', E', w)$ (right side) build from a P_3 $G = (V = \{a, b, c\}, E = \{z, y\})$ (left side). The weight function w for bold, normal and dashed edges are 2, 1 and 0 respectively.

First, consider an arbitrary independent set S of G . From S , we make a minimal edge cover F of G' of size at least $x(|S| + m)$. For each $e = uv \in E$, there is a vertex $f(e) \in ((V \setminus S) \cap \{u, v\})$ because S is an independent set. Let $X = \{f(e) : e \in E\}$, then choose arbitrarily a vertex $r \in X$. We set $F = \{f(e)v_{e_i} : e \in E, i \leq x\} \cup \{r'v' : v \in (V \setminus X)\}$. We deduce $\text{uec}(G', w) \geq w(F) = x(m + |S|)$ and considering S as a maximum independent set induces:

$$\text{uec}(G', w) \geq x(m + \alpha(G)) \quad (3.19)$$

Conversely, assume that F is a minimal edge cover of G' . We will polynomially modify F into another minimal edge cover F' such that $w(F') \geq w(F)$ and holds the following property.

Property 3.37. *We can assume that F satisfies the following facts:*

- (a) *for each $e = uv \in E$ at least one of u' or v' is a center of a star in F .*
- (b) *for each $e = uv \in E$, any vertex of V'_e is a leaf and its center is u' or v' .*

Remind that for an star $K_{1,1}$, both endpoints can be center and leaf.

Proof. For (a), suppose there exist some edges $e = uv \in E$ such that none of u' and v' is a center in F . We modify F into F'' by repeating the following process for all of such edges. We make a new star S' with center t as follows: if none of the centers of u' and v' are in V'_e , then $t = u$, otherwise t is one of u' or v' which its center is in V'_e . Let $S = \{ab \in F : a \in V'_e \cup \{t\}\}$ and $S' = \{tr : r \in V'_e\}$, then $F'' = (F \setminus S) \cup S'$. Since F is a minimal edge cover of G then F'' is a spanning star forest in G' with possibly some isolated vertices in V'_C . Considering the weight function w , $w(F'') \geq w(F)$. Easily by connecting all of the trivial stars in F'' to t , we make a new minimal edge cover F' such that $w(F') \geq w(F'') \geq w(F)$.

For (b), by considering (a), w.l.o.g. suppose u' is a center in F . Let $S = \{ab : a \in V'_e\}$ and $S' = \{ur : r \in V'_e\}$. Now $F'' = (F \setminus S') \cup S$ is a spanning star forest with possibly trivial stars of G' with $w(F'') \geq w(F)$ which satisfies (b). Notice after these stages, we may create of some isolated vertices included in V'_C . However, connecting every isolated vertices in V'_C to an arbitrary center in V'_C induces a minimal edge cover F' with larger weight. \square

Let $X' = \{v : vu \in F, u \in V'_E\}$ and $I' = V'_C \setminus X'$. By considering (a) in Property 3.37, $I = \{v : v' \in I'\}$ is an independent set of G . Since for each minimal edge cover F , there exist a minimal edge cover F' such that:

$$w(F) \leq w(F') = x(m + |I|) \leq x(m + \alpha(G)) \quad (3.20)$$

Hence by considering inequality (3.19) $\text{uec}(G', w) = x(m + \alpha(G))$.

Let F be a ρ -approximation solution for WEIGHTED UPPER EC for (G', w) and I be an independent set of G which made by F then:

$$\rho \leq \frac{w(F)}{\text{uec}(G', w)} \leq \frac{w(F')}{\text{uec}(G', w)} = \frac{x(m + |I|)}{x(m + \alpha(G))} = \frac{m + |I|}{m + \alpha(G)} \quad (3.21)$$

since G is connected of maximum degree Δ , we know $n \leq \Delta\alpha(G)$ (using Brook's Theorem), and then $m \leq \frac{\Delta^2}{2}\alpha(G)$. So, we can deduce that:

$$\begin{aligned} \Rightarrow 1 - \rho &\geq \frac{\alpha(G) - |I|}{m + \alpha(G)} \geq \frac{\alpha(G) - |I|}{(1 + \frac{\Delta^2}{2})\alpha(G)} \\ &\Rightarrow (1 + \frac{\Delta^2}{2})\rho - \frac{\Delta^2}{2} \leq \frac{|I|}{\alpha(G)} \end{aligned}$$

or equivalently $\frac{|I|}{\alpha(G)} \geq (1 + \frac{\Delta^2}{2}) \cdot \frac{w(F)}{\text{uec}(G', w)} - \frac{\Delta^2}{2}$. Remind that for the case $x = 1$, G' is a binary weighted graph and $w \in \{0, 1\}$ and $|V'| = n + m$. In the following property, we show that G' for the case that $x = 1$ is a c -dense graph with $c \geq \frac{4}{(2+\Delta)^2}$.

Property 3.38. *The graph G' when $x = 1$ is a c -dense graph with $c \leq \frac{4}{(2+\Delta)^2}$.*

Proof. Regarding the construction of $G' = (V', E', w)$, we have

$$N = |V'| = n + m \leq n + \frac{\Delta}{2}n = (\frac{2 + \Delta}{2})n. \quad (3.22)$$

On the other hand, since G' is a binary graph, then $\bar{w} = 1$ and we have $\sum_{u,v \in V'} w(u, v) = 2(\frac{n(n-1)}{2} + 2m) \geq n^2$. Using inequality (3.22), we have:

$$\sum_{u,v \in V'} w(u, v) \geq \frac{4}{(2 + \Delta)^2} N^2 \quad (3.23)$$

□

So G' is a c -dense graph with $c = \frac{4}{(2+\Delta)^2}$. Putting $\Delta = 3$ deduces $c = \frac{4}{25}$. □

He and Liang in [104], studied the MAX STAR FOREST problem in c -dense unweighted graphs where $c \in (0, 1)$. They have shown that MAX STAR FOREST problem in c -dense graphs does not admit a polynomial time approximation scheme (PTAS) unless $\mathbf{P} = \mathbf{NP}$, specifically, they proved that for any $c \in (0, 1)$, there exists $\varepsilon = \varepsilon(c) > 0$ such that approximating MAX STAR FOREST problem in c -dense graphs within a factor $(1 - \varepsilon)$ is NP-hard. Here, we strengthen the result by proving the existence of a constant lower bounds for WEIGHTED UPPER EC in c -dense graphs and particularly in c -dense k -trees for some different constants c .

Corollary 3.39. *WEIGHTED UPPER EC is not approximable within $\frac{1043}{1045} + \varepsilon$, $\frac{431}{432} + \varepsilon$, $\frac{620}{621} + \varepsilon$ and $\frac{835}{836} + \varepsilon$ for every $\varepsilon > 0$ unless $P=NP$ in the class of weighted c -dense k -trees with c equals to $\frac{4}{25}$, $\frac{1}{9}$, $\frac{4}{49}$ and $\frac{1}{16}$ respectively, even for binary weight function.*

Proof. By Property 3.38, we know that the construction proposed in Theorem 3.36, for the case $x = 1$ is a c -dense graph with $c = \frac{4}{(2+\Delta)^2}$. Thus, by Theorem 3.36, we deduce that a $(\frac{(1+\Delta^2)}{2}\rho - \frac{\Delta^2}{2})$ -approximation for MAX IS in graphs of maximum degree Δ is reachable in polynomial time from a ρ -approximation algorithm of WEIGHTED UPPER EC on c -dense k -trees with $c \leq \frac{4}{(2+\Delta)^2}$. On the other hand, in [51] it is proved MAX IS is not approximable within ratio $\frac{94}{95} + \varepsilon$, $\frac{47}{48} + \varepsilon$, $\frac{45}{46} + \varepsilon$ and $\frac{43}{44} + \varepsilon$ in graphs of maximum degree 3, 4, 5 and 6 respectively. Hence, by a simple calculation, we complete the proof. \square

By considering $x = 1$ and by deleting all edges of weight 0 in the graph G' given in Theorem 3.36 (let H be the resulting graph), we can infer that all the above bounds are valid for UPPER EC (or equivalently MAX STAR FOREST problem) in c -dense unweighted graphs.

Corollary 3.40. *UPPER EC is not approximable within $\frac{1043}{1045} + \varepsilon$, $\frac{431}{432} + \varepsilon$, $\frac{620}{621} + \varepsilon$ and $\frac{835}{836} + \varepsilon$ for every $\varepsilon > 0$ unless $P=NP$ in c -dense graphs with c equals to $\frac{4}{25}$, $\frac{1}{9}$, $\frac{4}{49}$ and $\frac{1}{16}$ respectively.*

Proof. By the Property 3.38, the construction $G' = (V', E', w)$ proposed in Theorem 3.36 for the case $x = 1$ is a binary weighted c -dense graph with $c \leq \frac{4}{(2+\Delta)^2}$. Hence, the graph $H = (V', E'')$ with $E'' = e \in E' : w(e) = 1$ is a c -dense unweighted graph. On the other hand, in the proof of Theorem 3.36, it is shown that any optimal solution of WEIGHTED UPPER EC F of G' is a subset of edges with weight 1. So, all the upper bounds of 3.39 is valid for UPPER EC. \square

He and Liang in [104], proposed an approximation algorithm for unweighted variant of spanning star forest problem in c -dense graph with ratio $0.804 + 0.196\sqrt{c}$. Note that this factor is larger than 0.835 when $c = \frac{4}{25}$. Nevertheless however, we showed that in Corollary 3.40, for $c = \frac{4}{25}$, it is hard to approximate MAX STAR FOREST problem within ratio $\frac{1043}{1045} + \varepsilon \approx 0.998$. Finally, note that UPPER EDGE COVER has at least a PTAS on everywhere- c -dense graphs using the result given in [103].

Positive approximation result

Now, we propose a positive approximation result of WEIGHTED UPPER EC via the use of an approximation preserving reduction from MAX STAR FOREST problem which polynomially transforms any ρ -approximation into a $\frac{k-1}{k+1}\rho$ -approximation for WEIGHTED UPPER EC.

Theorem 3.41. *In k -trees, WEIGHTED UPPER EC is $\frac{k-1}{2(k+1)}$ -approximable.*

Proof. The proof uses an approximation preserving reduction from MAX STAR FOREST problem which polynomially transforms any ρ -approximation into a $\frac{k-1}{k+1}\rho$ -approximation for WEIGHTED UPPER EC. Then, using the 0.5-approximation of MAX STAR FOREST problem given in [143], we will get the expected result.

Consider an edge-weighted k -tree (G, w) where $G = (V, E)$ and assume G is not complete. Let $\mathcal{S} = \{S_1, \dots, S_r\} \subseteq E$ be a nice spanning star forest of (G, w) (see Property 3.26) which is a ρ -approximation of MAX STAR FOREST problem. Let $opt_{\text{Max SFP}}$ denotes the value of optimal solution of MAX STAR FOREST problem, then we have:

$$w(\mathcal{S}) \geq \rho \cdot opt_{\text{Max SFP}}(G, w) \quad (3.24)$$

Now, we show how to modify \mathcal{S} into a minimal edge cover S without loosing "too much".

Before, we need to introduce some definitions and notations. A *vertex-coloring* $\mathcal{C} = (C_1, \dots, C_q)$ of a graph G is a partition of vertices into independent sets (called *colors*). The *chromatic number* of G , denoted $\chi(G)$, is the minimum number of colors used in a vertex-coloring. If G is a k -tree, it is well known that $\chi(G) = k + 1$ and such an optimal vertex-coloring can be done in linear time; hence, consider any optimal vertex-coloring $\mathcal{C} = \{C_1, \dots, C_{k+1}\}$ of G . Moreover, in k -trees we know that each vertex $u \in C_i$ of color i is adjacent to some vertex $v \in C_j$ of color j for every $j \neq i$.

We color the edges of $E(\mathcal{S})$ incident to every isolated vertices of Triv using the $k + 1$ colors where the color of such edge is given by the same color of its leaf. Formally, let $E' = \{uv \in E : v \in \text{Triv}\} \subseteq E(\mathcal{S})$ be the subset of edges incident to isolated vertices Triv and let $E_i = \{cv = e_v(\mathcal{S}) \in E(\mathcal{S}) : v \in C_i \setminus \text{Triv}\}$ for every $i \leq k + 1$ where c is some center of \mathcal{S} . The key property is the following:

Property 3.42. *for any $i < i'$, by deleting some edges of $E_i \cup E_{i'}$ and by adding edges from E' we obtain a minimal edge cover.*

Proof. It is valid because each vertex of color i is adjacent to some vertices of every other colors. Formally, fix two indices $1 \leq i < i' \leq k + 1$. Iteratively apply the following procedure: consider $v \in \text{Triv}$; there is $u \in V \setminus \text{Triv}$ such that $u \in C_i \cup C_{i'}$ (say C_i) and $vu \in E$. By hypothesis, u is a leaf of some ℓ -star S_r of \mathcal{S} . If at this stage $\ell \geq 2$, then add edge $uv \in E'$ and delete edge $uc \in E_i$ of color i ; otherwise $\ell = 1$ and we just add edge $uv \in E'$. At the end, we get a minimal edge cover. \square

Now, consider i_1, i_2 with $i_1 < i_2$ such that $w(E_{i_1} \cup E_{i_2}) = \min\{w(E_i \cup E_{i'}) : 1 \leq i < i' \leq k + 1\}$. Using Property 3.42 we can polynomially find a minimal edge cover S of (G, w) . By construction, $\sum_{i=1}^{k+1} w(E_i) \leq w(E(\mathcal{S}))$ and then:

$$w(E_{i_1} \cup E_{i_2}) \leq \frac{2}{k+1} w(E(\mathcal{S})) \quad (3.25)$$

Hence using inequalities (3.24) and (3.25), we get:

$$w(S') \geq w(E(\mathcal{S})) - w(E_{i_1} \cup E_{i_2}) \geq \frac{k-1}{k+1} w(E(\mathcal{S})) \geq \frac{k-1}{k+1} \rho \cdot opt_{\text{MaxWSSF}}(G, w)$$

Finally, since $opt_{\text{MaxWSSF}}(G, w) \geq \text{uec}(G, w)$ we get the expected result. \square

3.5.5 Weighted Upper EC in bounded degree graphs

In this section, we propose some positive approximation results for graphs of bounded degree in complement to those give in Corollary 3.32.

Theorem 3.43. *In general graphs with maximum degree Δ , there is an approximation preserving reduction from WEIGHTED UPPER EC to MAX EXT WEC with expansion $c(\rho) = \frac{1}{\Delta} \times \rho$.*

Proof. Consider an edge-weighted graph (G, w) of maximum degree $\Delta(G)$ bounded by Δ as an instance of WEIGHTED UPPER EC. We make an instance (G, w, U) of MAX EXT WEC by putting all pendant edges of G in the forced set U . Property 3.26 also works in this context since U is the set of pendant edges. Let opt_{EWEC} and opt_{WUEC} be the values of optimal solutions of MAX EXT WEC and WEIGHTED UPPER EC respectively. In particular, we deduce $opt_{EWEC}(G, w, U) \geq opt_{WUEC}(G, w)$ because, U belongs to any minimal edge cover. Let $\mathcal{S} = \{S_1, \dots, S_r\} \subseteq E$ be a nice spanning star forest of (G, w) containing U satisfying:

$$w(\mathcal{S}) \geq \rho \times opt_{EWEC}(G, w, U) \geq \rho \times opt_{WUEC}(G, w) \quad (3.26)$$

For each $t \in \text{Triv}$, we choose two edges incident to it with maximum weights $e_1^t = tx_t$ and $e_2^t = ty_t$ in $E \setminus E(\mathcal{S})$ (since by construction $d_G(v) \geq 2$), i.e. $w(e_1^t) \geq w(e_2^t) \geq w(tv)$ for all possible v ; let $W = \sum_{t \in \text{Triv}} (w(e_1^t) + w(e_2^t))$ be this global quantity. Also, recall that V_c and V_l are the set of vertices labeled by centers and leaves respectively according to \mathcal{S} . We build a new vertex weighted graph $G(\mathcal{S}) = G' = (V', E', w')$ with maximum degree $\Delta(G') \leq \Delta(G) - 1$ as follows:

- $V' = V_l$.
- $uv \in E'$ if and only if there exists $t \in \text{Triv}$ with $tx_t = tu$ and $ty_t = tv$.
- For $v \in V'$, we set $w'(v) = w(e_v(\mathcal{S}))$ ⁴.

Clearly, G' is a graph with bounded degree $\Delta - 1$. We mainly prove that from any independent set $I \subseteq V'$ we can polynomially build an upper edge cover S_I of G satisfying:

$$w(S_I) \geq w'(I) + \left(W - \sum_{t \in \text{Triv}} w(e_1^t) \right) \geq w'(I) \quad (3.27)$$

Let $I \subseteq V'$ be a maximal independent set of G' . This implies $V' \setminus I$ is a vertex cover of G' . By construction of G' , for every $t \in \text{Triv}$, at least one vertex x_t or y_t is not in I (say x_t in the worst case). Recall $e_{x_t}(\mathcal{S})$ is the edge of spanning star forest incident to x_t (since $x_t \in V_l$). We will iteratively apply the following procedure for all $t \in \text{Triv}$ to build S_I : if the current ℓ -star S_r of \mathcal{S} containing $e_{x_t}(\mathcal{S})$ satisfies $\ell \geq 2$ (it is true initially by hypothesis), then delete edge $e_{x_t}(\mathcal{S})$ from \mathcal{S} , add edge e_1^t and update spanning star forest \mathcal{S} . Otherwise, $\ell = 1$ and only add e_1^t . At the end of the procedure, we get a minimal edge cover S_I of G satisfying inequality (3.27).

⁴We recall $e_v(\mathcal{S})$ is the edge of \mathcal{S} linking leaf v to its center.

Now, in order to propose a feasible solution of MAX IS for G' , apply a greedy algorithm: at each step, taking one vertex with maximum weight w' and removing all the remaining neighbors of it. Thus, we have:

$$w'(I) \geq \frac{w'(V')}{\Delta(G') + 1} \geq \frac{w(\mathcal{S})}{\Delta(G)} \quad (3.28)$$

Hence, using inequalities (3.26), (3.27) and (3.28), we get the expected result. \square

Using the 0.5-approximation of MAX EXT WEC presented in Section 3.4, we deduce:

Corollary 3.44. *WEIGHTED UPPER EC is $\frac{1}{2\Delta}$ -approximable in graphs with bounded degree Δ .*

Now, assume that the weight function w just contains 1, 2 then for every $t \in \text{Triv}$, removing $e_{x_t}(\mathcal{S})$ from \mathcal{S} and adding tx_t to it, gives a minimal edge cover \mathcal{S}_I of G which satisfies:

$$w(\mathcal{S}_I) \geq w(\mathcal{S}) - ((2 - 1)(V' \setminus I)) \geq \frac{1}{2}w(\mathcal{S}) \quad (3.29)$$

So for general graphs with $w \in \{1, 2\}$, we can easily deduce:

Corollary 3.45. *WEIGHTED UPPER EC in graphs with $w \in \{1, 2\}$ is $\frac{1}{4}$ -approximable.*

3.6 Conclusion

In this chapter, we considered the extension version of weighted edge cover problem. Both maximization and minimization versions of the problem are studied. We showed that for a given graph $G = (V, E)$ and a forced set $U \subseteq E$, even deciding the existence of a minimal edge cover containing U is a NP-hard problem. However, by introducing a dynamic programming algorithm for bounded tree-width graphs, we show that not only the decision version is decidable in polynomial time but also both MIN EXT WEC and MAX EXT WEC are solvable in polynomial time. Since the decision variants of the problems are NP-hard, we just focus on the complete weighted graph to ensure that finding some feasible solutions can be done in polynomial time (if any).

| | <i>w-general</i> | <i>c-relaxed</i> | <i>extended c-relaxed</i> |
|--------------------|--|--|---|
| MIN EXT WEC | NP-hard non-apx at all FPT w.r.t. $ U $ | NP-hard $c > 1$ poly-time $\frac{1}{2} \leq c \leq 1$ c apx-ratio $\frac{7+c}{8} - \epsilon$ in-apx | NP-hard $c > 1$ poly-time $c = 1$ $\frac{c+1}{2}$ apx-ratio $\frac{7+c}{8} - \epsilon$ non-apx |
| MAX EXT WEC | 0.5 apx-ratio $\frac{7}{8} + \epsilon$ in-apx | — | — |

Table 3.1: The results given for complete graphs

Moreover, we have studied the problem in the framework of approximability. Table 3.1 summarizes the results obtained for the two optimization problems on complete graphs.

Further, we have proposed some positive and negative approximability results of WEIGHTED UPPER EC for special classes of graphs as well as bounded degree graphs. This problem is a particular case of MAX EXT WEC, when the forced set $U = \emptyset$.

We finish this chapter with a remark regarding the complexity of WEIGHTED UPPER EC with respect to the tree-width. We showed that MAX EXT EC is solvable in time $O^*(6^k)$ for graphs with tree-width bounded by k . So, WEIGHTED UPPER EC is solvable in $O^*(6^k)$ too. On the other hand, Lokshantov et al. proved in [132] that, assuming *SETH*, there is no $O^*((3 - \epsilon)^k)$ algorithm with $\epsilon > 0$ for MIN DS. It means that, assuming *SETH*, UPPER EC does not admit any $O^*((3 - \epsilon)^k)$ algorithm. Since we focused on weighted version of the problem, it would be interesting to reduce this gap.

Extension of Some Vertex Graph Problems

Contents

| | | |
|------------|--|------------|
| 4.1 | Introduction | 102 |
| 4.1.1 | Related work | 102 |
| 4.1.2 | Contributions | 104 |
| 4.2 | EXT VC, EXT IS | 104 |
| 4.2.1 | Computational complexity | 106 |
| 4.2.2 | Parameterized complexity | 113 |
| 4.2.3 | Price of extension | 115 |
| 4.2.4 | Generalizations to extensions of H -graph cover and H -free subgraph | 118 |
| 4.3 | EXT CVC and EXT NSIS | 120 |
| 4.3.1 | Computational complexity | 121 |
| 4.3.2 | Price of extension | 126 |
| 4.3.3 | Upper CVC | 130 |
| 4.4 | EXT DS | 133 |
| 4.4.1 | Computational complexity | 133 |
| 4.4.2 | Parameterized complexity | 137 |
| 4.4.3 | Price of extension | 138 |
| 4.5 | Conclusion | 139 |

Most of the results presented in this chapter appeared in the following conference papers:

- Extension of vertex cover and independent set in some classes of graphs. In Proceedings of the 11th International Conference on Algorithms and Complexity, CIAC 2019. Joint work with Katrin Casel, Henning Fernau, Jerome Monnot and Florian Sikora.

- Extension and its price for the connected vertex cover problem. In Proceedings of the 30th International Workshop on Combinatorial Algorithms, IWOCA 2019. Joint work with Nikolaos Melissinos, Jerome Monnot and Aris Pagourtzis.

4.1 Introduction

In this chapter, we consider *extension problems* related to several vertex optimization problems in graphs, namely VERTEX COVER, INDEPENDENT SET, CONNECTED VERTEX COVER, NON SEPARATING INDEPENDENT SET and DOMINATING SET. Some of these problem are already introduced in Chapter 1. A subset $S \subseteq V$ of a given connected graph $G = (V, E)$ is a *connected vertex cover* (resp. a *non separating independent set*) if S is a vertex cover and the subgraph $G[S]$ induced by S is connected (resp. if S is an independent set and it is not a cut-set of G). The goal of the optimization variant of the two former problems is to find a connected vertex cover (resp. a non separating independent set) of minimum (resp. maximum) size. Informally, for a hereditary (resp. anti-hereditary) vertex graph problem Π , the noticed decision problem denoted by EXT Π , in which a pair (G, U) is given as an input where $G = (V, E)$ and $U \subseteq V$, and it is asked if there is a maximal (resp. minimal) solution S such that $S \subseteq U$ (resp. $U \subseteq S$). Regarding this decision problem, we obtain some NP-hardness results for all the mentioned problems even for restricted instances to bipartite or planar graphs. Most of our NP-hardness results translate into ETH-hardness results as well. Moreover, we consider the problems in parameterized complexity point of view with respect to the size of forced set or forbidden set¹ and their dual. Furthermore, we study the *Price of Extension* (PoE), a measure that reflects the distance of a vertex set U to its maximum efficiently computable subset that is extensible to a minimal/maximal solution, and provide negative and positive results from approximation point of view for PoE in general and special graphs.

4.1.1 Related work

The extension variants of some classical graph problem have been studied in the literature [32, 17, 16]. In [17], an NP-hardness proof for the extension of minimal dominating set (denoted by EXT DS here), when the instance is restricted to planar cubic graphs was presented. Similarly, it was shown in [16] that extension of minimal vertex cover (EXT VC) is NP-hard, even restricted to planar cubic graphs, which leads to an NP-hardness proof of EXT IS in our consideration. Further, in order to enumerate all (inclusion-wise) minimal dominating sets of a given graph, Kanté et al. studied an extension variant of dominating set, in which, for a given graph G and two subsets of vertices X, Y , the goal is to find all minimal dominating set containing X but excluding Y (In our setting of EXT DS, $Y = \emptyset$). They proved that EXT DS is NP-complete, even in special graph classes like split graphs, chordal graphs and line graphs [117, 116]. Moreover, they proposed a linear algorithm for split graphs when X, Y is a partition of the clique part [115].

¹we recall that forbidden set is defined as \overline{U} , i.e. $V \setminus U$ in our context.

MIN VC and MAX IS are two classical graph optimization problems which have been studied in many researches. The problems are NP-hard in planar graphs [88], but they are polynomial solvable for several graph classes like bipartite graphs, chordal graphs and graphs with bounded tree-width [89, 35]. From (in-)approximability point of view, MIN VC is known to be APX-complete in cubic graphs [5], 2-approximable in general graphs [12, 106] and admits a polynomial approximation scheme in planar graphs [15] and it is proved that MAX IS in general graphs is not approximable within ratio $n^{\varepsilon-1}$ for any $\varepsilon > 0$ unless $P = NP$ [165], while it is APX-complete in bounded degree graphs [154].

Garey and Johnson proved that *minimum connected vertex cover* problem denoted by MIN CVC is NP-hard in planar graphs of maximum degree 4 [87]. Moreover, it is shown in [153, 156] that the problem is polynomially solvable for graphs of maximum degree 3, while NP-hardness proofs for bipartite and for bi-connected planar graphs of maximum degree 4, are presented in [74, 83, 148]. The approximability of MIN CVC has been considered in some more recent studies. The NP-hardness of approximating MIN CVC within $10\sqrt{5} - 21$ is proven in [83] while a 2-approximation algorithm is presented in [149]. Moreover, in [74] the problem is proven APX-complete in bipartite graphs of maximum degree 4. They also propose a $\frac{5}{3}$ -approximation algorithm for MIN CVC for any class of graphs where MIN VC is polynomial-time solvable. Parameterized complexity for MIN CVC and *maximum non separating independent set* denoted by MAX NSIS have been studied in [126, 130] while the enumeration of minimal connected vertex covers is investigated in [94] where it is shown that the number of minimal connected vertex covers of a graph of n vertices is at most 1.8668^n , and these sets can be enumerated in time $O(1.8668^n)$. For chordal graphs (even for chordality at most 5), the authors are able to give a better upper bound. Finally, one can find problems that are quite related to MIN CVC in [39]. To the best of our knowledge, the extension variant of connected vertex cover has not been studied in the literature.

Max-Min or Min-Max optimization variants have been studied for many classical graph problems from the (in-)approximability point of view in recent years. For example, in Manlove thesis [136] a framework and a systematic studies have been proposed. Dealing with UPPER VC in [30, 141], a tight approximation result in general graphs is provided: a $n^{-\frac{1}{2}}$ approximation algorithm together with a proof for in-approximability of the problem within factor $n^{\varepsilon-\frac{1}{2}}$, for any $\varepsilon > 0$. The MIN-MAX IS problem or equivalently *min independent dominating set* denoted by MIN ISDS asks, given a graph $G = (V, E)$, for a subset $S \subseteq V$ of minimum size that is simultaneously independent and dominating. From the NP-hardness and exact solvability point of views, MIN ISDS is equivalent to UPPER VC [137], but they seem to behave differently in terms of approximability and parameterized complexity [12]. Although MIN ISDS is polynomially solvable in strongly chordal graphs [77], it is hard to approximate within $n^{\varepsilon-1}$, for any $\varepsilon > 0$, in certain graph classes [77, 61]. UPPER DS is considered from the approximation perspective [17], where it is shown the problem does not admit an $n^{\varepsilon-1}$ approximation for any $\varepsilon > 0$, unless $P = NP$. It is also proved that UPPER DS remains APX-hard on cubic graphs. The authors complemented their results by proposing an APX-algorithm for bounded degree graphs, as well as an EPTAS on planar graphs.

4.1.2 Contributions

We study the associated decision problem to extension problems for some vertex graph problems in this chapter. We prove that the EXT Π for any vertex problem Π which is mentioned above, is NP-complete in general, even for some restricted cases. We also study the parameterised complexity of these problems with respect to parameters related to the pre-solution as well as the optimality of simple exact algorithms under the Exponential-Time Hypothesis (ETH). We further focus on bounds for PoE, providing inapproximability results for new natural optimization problems corresponding to the decision versions.

We organize our results in this chapter as follows. In Section 4.2, the extension variant of two classic graph problems VERTEX COVER, INDEPENDENT SET is studied. By proposing a characterization of a *yes*-instance of EXT VC, we show that the two above problems are equivalent. Then, we prove that the problems are NP-complete to decide even for sub-cubic planar bipartite graphs while they are polynomial-time decidable in chordal graphs, circular-arc graphs and trees. We also study parameterized complexity of these problems in this section and comment on lower bound results based on ETH. Concerning the new concept *price of extension*, we introduce two optimization problems and discuss (in-)approximability for the according optimization problems. Finally, we generalize our results to H -free graphs for some fixed H .

In Section 4.3, we focus on the extension variant of two problems CONNECTED VERTEX COVER and NON SEPARATING INDEPENDENT SET. After showing the relation between EXT VC and EXT CVC, we provide additional hardness results for EXT CVC in bipartite graphs and weakly triangulated graphs, the latter leading to hardness results for UPPER VC and UPPER CVC. We then focus on bounds for PoE, providing in-approximability results for EXT_{max}CVC in general and bipartite graphs. Finally, we discuss the (in-)approximability of a special case of EXT_{max}CVC, namely UPPER CVC in this chapter.

We study the extension variant of DOMINATING SET in Section 4.4. We prove the EXT DS is NP-hard in planar bipartite graphs of maximum degree 3. Moreover, we show that EXT DS is W[3]-complete with respect to the size of forced set, as we show that the problem is equivalent to the extension variant of HITTING SET in parameterized complexity point of view. We finally focus on bounds for PoE, providing in-approximability results for EXT_{max}DS in general and bipartite graphs as well as graphs with bounded degree.

4.2 EXT VC, EXT IS

In this section, we consider extension variants of the classical graph problems VERTEX COVER and INDEPENDENT SET. For a given graph $G = (V, E)$, a subset $S \subseteq V$ is called vertex cover (resp. independent set) if each edge $e \in E$ is incident to at least one vertex $v \in S$ (resp. if $G[S]$ contains no edges.) Here, for a given graph $G = (V, E)$ and a vertex set $U \subseteq V$, it is asked if there exists a *minimal* vertex cover (resp. *maximal* independent set) S with $U \subseteq S$ (resp. $U \supseteq S$). These problems formally defined as follows:

EXT VC

Input: A graph $G = (V, E)$ together with a forced set $U \subseteq V$.

Question: Is there a minimal vertex cover $S \subseteq V$ such that $U \subseteq S$?

EXT IS

Input: A graph $G = (V, E)$ together with a co-forbidden set $U \subseteq V$.

Question: Is there a maximal independent set $S \subseteq V$ such that $U \supseteq S$?

Recall that, in EXT VC (resp. EXT IS), the set U is also referred to as the set of *forced* (resp. *co-forbidden*) vertices. Since complements of a maximal independent set is a minimal vertex cover, we can conclude:

Remark 4.1. (G, U) is a yes-instance of EXT VC if and only if $(G, V \setminus U)$ is a yes-instance of EXT IS.

Since adding or deleting edges between vertices of U does not change the minimality of feasible solutions of EXT VC, we can first state the following.

Remark 4.2. For EXT VC (and for EXT IS) one can always assume $G[U]$ (resp. $G[V \setminus U]$) is either a clique or an independent set.

The following theorem gives a combinatorial characterization of yes-instances of EXT VC that is quite important in our subsequent discussions.

Theorem 4.3. Let $G = (V, E)$ be a graph and $U \subseteq V$ be a set of vertices. The three following conditions are equivalent:

- (i) (G, U) is a yes-instance of Ext VC.
- (ii) $(G[N_G[U]], N_G[U] \setminus U)$ is a yes-instance of Ext IS.
- (iii) There exists an independent dominating set $S' \subseteq N_G[U] \setminus U$ of $G[N_G[U]]$.

Proof. In the following arguments, let $G = (V, E)$ be a graph. Let us first look at conditions (ii) and (iii). By our previous discussions, condition (ii) is equivalent to: $(G[N_G[U]], U)$ is a yes-instance of EXT VC. Assume there is a minimal vertex cover S of $G[N_G[U]]$ with $U \subseteq S$. Hence, in particular we deduce $N_G[v] \not\subseteq U$ for every $v \in U$ by minimality of S . Condition (ii) therefore entails the existence of an independent set S' of $G[N_G[U]]$ with $S' \subseteq (N_G[U] \setminus U)$ and $U \subseteq N_G[S']$. Hence, condition (ii) implies condition (iii). Conversely, let $S' \subseteq N_G[U] \setminus U$ be an independent dominating set of $G[N_G[U]]$. Clearly, $S = N_G[U] \setminus S'$ is a vertex cover of $G[N_G[U]]$. If S were not minimal, then there would be a vertex $v \in S$ with $N_{G[N_G[U]]}(v) \subseteq S$, as then v would not possess a private edge. But then v would not be dominated by any vertex from S' , violating the assumption that S' is a dominating set of $G[N_G[U]]$. Hence, conditions (ii) and (iii) are equivalent.

Now, we will prove the equivalence between items (i) and (iii). Let S be a minimal vertex cover of G with $U \subseteq S$. Clearly, $S \cap N_G[U]$ is a vertex cover of $G[N_G[U]]$, but notice that it need

not be minimal, as private edges of $v \in S \cap N_G[U]$ need not lie in the graph induced by $N_G[U]$. The set $S' = (V \setminus S) \cap N_G[U] \subseteq N_G[U] \setminus U$ is an independent set (as the complement of $S \cap N_G[U]$ within $G[N_G[U]]$) which dominates all the vertices in U . Namely, consider any $u \in U$ and assume that $u \notin N_G[S']$. Then, $N_G[u] \subseteq S$, contradicting minimality of S . We turn S' into a maximal independent set of the induced graph $G[N_G[U] \setminus U]$, by adding some vertices from $N_G[U] \setminus U$ to S' . Observe that the resulting set S'' is also a maximal independent set in $G[N_G[U]]$ and hence satisfies condition (iii), because each $u \in U$ has a private edge (as being part of the minimal vertex cover S of G , connecting u to some $v \in S'$). Conversely, assume the existence of an independent dominating set R of G satisfying (iii). Hence, R is an independent set with $R \subseteq (N_G[U] \setminus U)$ and $U \subseteq N_G[R]$. Let X be any maximal independent set of $G[V \setminus N_G[R]]$, for instance, produced by some greedy procedure. Let $S' := R \cup X$. By construction, S' is an independent set in G . If S' were not maximal, then we would find some $x \in S'$ with $N_G(x) \cap S' = \emptyset$. Clearly, $x \notin N_G[R]$. But as x has no neighbors in X , it could have been added to X by the mentioned greedy procedure. In conclusion, S' is a maximal independent set. Hence, $S = V \setminus S'$ satisfies the condition (i). \square

4.2.1 Computational complexity

Here, we study hardness of EXT VC and EXT IS in some different graph classes. Due to Remark 4.1, two extension problems are equivalent in (in)-tractability point of view, so in each proof, we just focus on one of the problems.

EXT VC and EXT IS in bipartite graphs

In this section, we deal with bipartite graphs. In the following, We will show that EXT VC is NP-complete, even if restricted to cubic, or planar subcubic graphs. We add some algorithmic notes on planar graphs that are also valid for the non-bipartite case.

Theorem 4.4. *EXT VC is NP-complete in cubic bipartite graphs.*

Proof. We make a reduction from (3, B2)-SAT to EXT VC. For an instance $I = (\mathcal{C}, \mathcal{X})$ of (3, B2)-SAT where $\mathcal{C} = \{c_1, \dots, c_m\}$ and $\mathcal{X} = \{x_1, \dots, x_n\}$ are the set of clauses and variables respectively, we build a cubic bipartite graph $G = (V, E)$ by duplicating instance I as follows:

- make two vertices c_i, c'_i for each clause $c_i \in \mathcal{C}$.
- make two paths, $\langle x_i, l_i, m_i, r_i, \neg x_i \rangle$ and $\langle x'_i, l'_i, m'_i, r'_i, \neg x'_i \rangle$ by adding 10 new vertices for each variable $x_i \in \mathcal{X}$.
- add 3 new edges $l_i l'_i, m_i m'_i$ and $r_i r'_i$ for $i = 1, \dots, n$.
- for each variable x_i appears positively (resp. negatively) in each clause c_j , add two new edges $x_i c_j, x'_i c'_j$ (resp. $\neg x_i c_j, \neg x'_i c'_j$). These new edges, called *crossing edge*.

The construction is illustrated in Figure 4.1 and clearly, G is a cubic bipartite graph. Finally we set $U = \{c_i, c'_i : i = 1, \dots, m\} \cup \{m_j, m'_j : j = 1, \dots, n\}$. We claim that I is satisfiable if and only if G admits a minimal vertex cover containing U .

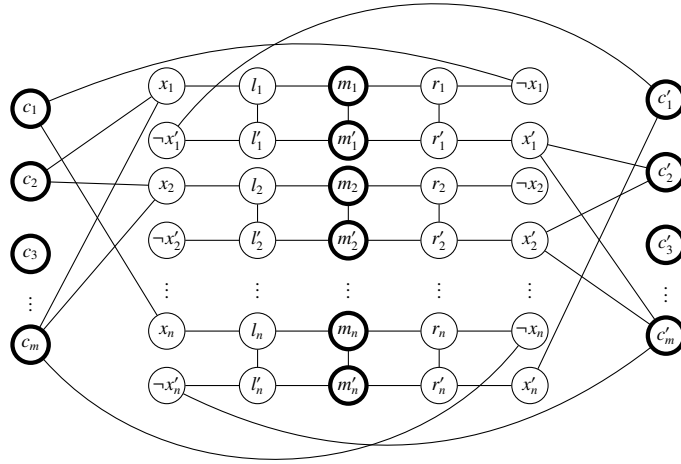


Figure 4.1: Graph $G = (V, E)$ for EXT VC built from I . Vertices of U have a bold border.

Assume I is satisfiable and let T be a truth assignment which satisfies all clauses. We set $S = \{\neg x_i, l_i, \neg x'_i, r'_i : T(x_i) = \text{true}\} \cup \{x_i, r_i, x'_i, l'_i : T(x_i) = \text{false}\} \cup U$. We can easily check that S is a vertex cover containing U . By the construction, it is easy to see that every vertex in $S \setminus (C \cup C')$ covers at least one private edge of G . So, for minimality, we just need to show that for any c_i (resp. c'_i , $1 \leq i \leq m$), there is at least one crossing edge which is covered by c_i (resp. c'_i). If not, then there is a vertex c_i or c'_i which all of the three crossing edges incident to it are covered by other vertices in S , hence, considering the assignment T , the clause c_i is not satisfied. Therefore, S is a minimal vertex cover.

Conversely, assume that G possesses a minimal vertex cover S containing U . For a fixed i , we know that $|\{l_i, l'_i, r_i, r'_i\} \cap S| \geq 2$ to cover the edges $l_i l'_i$ and $r_i r'_i$. If $\{l_i, r_i\} \subseteq S$ (resp. $\{l'_i, r'_i\} \subseteq S$), then S is not a minimal vertex cover because m_i (resp. m'_i) can be deleted, a contradiction. If $\{l_i, l'_i\} \subseteq S$ (resp. $\{r_i, r'_i\} \subseteq S$), then S must contain another vertex to cover $r_i r'_i$ (resp. $l_i l'_i$), leading to the previous case, a contradiction. Hence, if $\{l_i, r'_i\} \subseteq S$ (resp. $\{r_i, l'_i\} \subseteq S$), then $\{\neg x_i, \neg x'_i\} \subseteq S$ (resp. $\{x_i, x'_i\} \subseteq S$), since the edges $l'_i \neg x'_i$ and $r_i \neg x_i$ (resp. $l_i x_i$ and $r'_i x_i$) must be covered. Finally, by setting $T(x_i) = \text{true}$ if $\neg x_i \in S$ and $T(x_i) = \text{false}$ if $x_i \in S$ we obtain a truth assignment T which satisfies all clauses. The assignment T is a satisfying truth assignment follows by minimality rather than by the fact that any particular edge needs to be covered. \square

Corollary 4.5. *EXT IS is NP-complete in cubic bipartite graphs.*

In the following, we discuss restriction to planar graphs. In order to prove our results, we will present a new reduction from 4P3C3SAT, which is already explained in the beginning of this chapter.

Theorem 4.6. *EXT IS is NP-complete on planar bipartite subcubic graphs.*

Proof. The proof is based on a reduction from 4P3C3SAT. Let $I = (\mathcal{C}, \mathcal{X})$ be an instance of 4P3C3SAT, where $\mathcal{C} = \{c_1, \dots, c_m\}$ and $\mathcal{X} = \{x_1, \dots, x_n\}$ are the set of clauses and variables of I

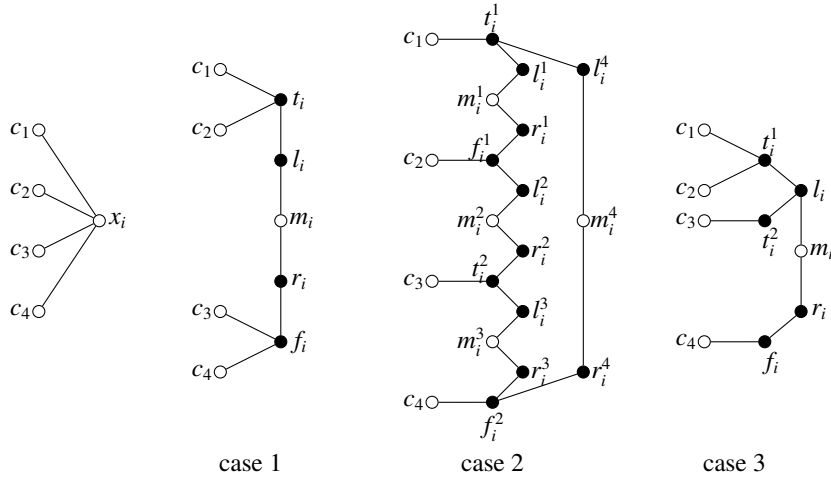


Figure 4.2: Construction of Theorem 4.6. On the left: A variable x_i appearing in four clauses c_1, c_2, c_3, c_4 in I . On the right, cases 1, 2, 3: The gadgets $H(x_i)$ in the constructed instance, depending on how x_i appears (negative or positive) in the four clauses. Black vertices denote elements of U .

respectively. W.l.o.g. suppose that variable x_i appears in clauses c_1, c_2, c_3, c_4 of instance I such that in the induced (embedded) subgraph $G_i = G[\{x_i, c_1, c_2, c_3, c_4\}]$ of the graph BP defined in Definition 1.1, $c_1x_i, c_2x_i, c_3x_i, c_4x_i$ is an anti-clockwise ordering of edges around x_i . By looking at G_i and considering x_i appears positively and negatively, the construction should satisfy one of the following cases:

- case 1: $x_i \in c_1, c_2$ and $\neg x_i \in c_3, c_4$;
- case 2: $x_i \in c_1, c_3$ and $\neg x_i \in c_2, c_4$;
- case 3: $x_i \in c_1, c_2, c_3$ and $\neg x_i \in c_4$.

Note that all other cases are included in these by rotations or replacing x_i with $\neg x_i$ or vice versa.

We start from the associated graph $BP = (X \cup C, E(BP))$, and build a planar bipartite graph H by replacing every node x_i in BP with one of the three gadgets $H(x_i)$ which are depicted in Figure 4.2. Let

$$F_1 = \{m_i : H(x_i) \text{ complies with cases 1 or 3}\}$$

and let

$$F_2 = \{m_i^1, m_i^2, m_i^3, m_i^4 : H(x_i) \text{ complies with case 2}\}.$$

The co-forbidden set $U = V(H) \setminus (F_1 \cup F_2 \cup C)$, where $C = \{c_i : 1 \leq i \leq m\}$. This construction is polynomial-time computable and H is a planar bipartite subcubic graph. We claim that H has a maximal independent set which contains only vertices from U if and only if I is satisfiable.

If T is a truth assignment of I which satisfies all clauses, then depending on $T(x_i) = \text{true}$ or $T(x_i) = \text{false}$, we define the independent set S_i corresponding to three different variable gadgets $H(x_i)$ as follows:

$$S_i := \begin{cases} \{t_i, r_i\} & \text{if } H(x_i) \text{ adapts to case 1 and } T(x_i) = \text{true,} \\ \{t_i^1, r_i^1, l_i^2, t_i^2, r_i^3, r_i^4\} & \text{if } H(x_i) \text{ adapts to case 2 and } T(x_i) = \text{true,} \\ \{t_i^1, t_i^2, r_i\} & \text{if } H(x_i) \text{ adapts to case 3 and } T(x_i) = \text{true,} \\ \{f_i, l_i\} & \text{if } H(x_i) \text{ adapts to case 1 and } T(x_i) = \text{false,} \\ \{l_i^1, f_i^1, r_i^2, l_i^3, f_i^2, l_i^4\} & \text{if } H(x_i) \text{ adapts to case 2 and } T(x_i) = \text{false,} \\ \{l_i, f_i\} & \text{if } H(x_i) \text{ adapts to case 3 and } T(x_i) = \text{false.} \end{cases}$$

We can see that $S = \bigcup_{1 \leq i \leq n} S_i$ is a maximal independent set of H which contains only vertices of U .

Conversely, suppose $S \subseteq U$ is a maximal independent set of H . By using maximality of S , we define an assignment T for I depending on different types of variable gadgets of H as follows:

- for case 1, one of l_i, r_i must be in S , hence we set $T(x_i) = \text{true}$ (resp. $T(x_i) = \text{false}$) if $r_i \in S$ (resp. $l_i \in S$).
- for case 2, at least one of vertices in each pair $\{(l_i^j, r_i^j) : 1 \leq j \leq 4\}$ must be in S . Hence, at most one of $(S \cap \{t_i^1, t_i^2\}) \neq \emptyset$ and $(S \cap \{f_i^1, f_i^2\}) \neq \emptyset$ is true. Thus we set $T(x_i) = \text{true}$ (resp. $T(x_i) = \text{false}$) if $(S \cap \{t_i^1, t_i^2\}) \neq \emptyset$ (resp. $(S \cap \{f_i^1, f_i^2\}) \neq \emptyset$).
- for case 3, one can see, similar to the previous two cases: if one of t_i^1, t_i^2 (resp. f_i) is in S , then none of f_i (resp. t_i^1, t_i^2) are in S , then we set $T(x_i) = \text{true}$ (resp. $T(x_i) = \text{false}$) if $(S \cap \{t_i^1, t_i^2\}) \neq \emptyset$ (resp. $f_i \in S$).

We obtain a valid assignment T . This assignment satisfies all clauses of I , since for all $c_j \in C$, $(N(c_j) \cap S) \neq \emptyset$ (by maximality of S). \square

Corollary 4.7. *Ext VC is NP-complete in sub-cubic planar bipartite graphs.*

Ext VC and Ext IS in chordal and circular-arc graphs

An undirected graph $G = (V, E)$ is *chordal* if and only if each cycle of G with a length at least four has a chord (an edge linking two non-consecutive vertices of the cycle) and G is *circular-arc* if it is the intersection graph of a collection of n arcs around a circle. We will need the following problem definition.

MINIMUM INDEPENDENT DOMINATING SET (MIN ISDS for short)

Input: A graph $G = (V, E)$.

Solution: Subset of vertices $S \subseteq V$ which is independent and dominating.

Output: Solution S that minimizes $|S|$.

WEIGHTED MINIMUM INDEPENDENT DOMINATING SET (OR WMIN ISDS for short) corresponds to the vertex-weighted variant of MIN ISDS, where each vertex $v \in V$ has a non-negative weight $w(v) \geq 0$ associated to it and the goal consists in minimizing $w(S) = \sum_{v \in S} w(v)$. If $w(v) \in \{a, b\}$ with $0 \leq a < b$, the weights are called *bivariate*, and $a = 0$ and $b = 1$ corresponds to *binary weights*.

Remark 4.8. MIN ISDS for chordal graphs has been studied in [76], where it is shown that the restriction to binary weights is solvable in polynomial-time. Bivalued MIN ISDS with $a > 0$ however is already NP-hard on chordal graphs, see [46]. WMIN ISDS (without any restriction on the number of distinct weights) is also polynomial-time solvable in circular-arc graphs [47].

Using the mentioned polynomial-time result of BINARY INDEPENDENT DOMINATING SET ON chordal graphs [76] and circular-arc graphs [47], we deduce:

Corollary 4.9. *EXT VC is polynomial-time decidable in chordal and in circular-arc graphs.*

Proof. By Remark 4.8, we can find, within polynomial-time, an independent dominating set S^* minimizing $w(S^*) = \sum_{v \in S^*} w(v)$ among the independent dominating sets of a weighted chordal graph or circular-arc graph (G, w) where $G = (V, E)$ and $\forall v \in V, w(v) \in \{0, 1\}$.

Let (G, U) be an instance of EXT VC where $G = (V, E)$ is a chordal graph (resp. a circular-arc graph). We will apply the result of [76] (resp. [47]) for (G', w) , where G' is the subgraph of G induced by $N_G[U]$ and $w(v) = 1$ if $v \in U$ and $w(v) = 0$ for $v \in N_G[U] \setminus U$. Obviously, (G', w) is a binary-weighted chordal graph (resp. circular-arc graph). So, an optimal independent dominating set S^* of (G', w) has a weight 0 if and only if $S^* \subseteq N_G[U] \setminus U$ is a maximal independent set of G' , otherwise $w(S^*) \geq 1$. Using Theorem 4.3, the result follows. \square

Corollary 4.10. *EXT IS are polynomial-time decidable in chordal and in circular-arc graphs.*

Farber's algorithm [76] (used in Corollary 4.10) runs in linear-time and is based on the resolution of a linear programming using primal and dual programs. Yet, it would be nice to find a (direct) combinatorial linear-time algorithm for chordal and circular-arc graphs, as this is quite common in that area. In the next section, we give a first step in this direction by presenting a characterization of yes-instances of EXT VC on trees.

EXT VC and EXT IS in trees

In this subsection, for any instance (G, U) of EXT VC, we assume that the forced vertex set U is an independent set because if a graph is a forest, after applying Remark 4.2, the new graph remains a forest.

Given an undirected tree $T_r = (V, E)$, where $r \in V$ is a specified vertex called *root*, we denote by $\text{fa}_{T_r}^i(v)$ of $v \in V$ for $i \geq 1$ the i -th visited vertex different from v in the unique path from v to r in T_r . For instance, $\text{fa}_{T_r}^1(v)$ is called *father* of v while $\text{fa}_{T_r}^2(v)$ is called *grandfather* of v . The set $\text{ch}_{T_r}(v) = N_T(v) \setminus \{\text{fa}_{T_r}^1(v)\}$ is called the set of *children* of v . The root has no father and its neighbors are its children. A *leaf* of T is a vertex v without children, i.e. $N_{T_r}(v) = \{\text{fa}_{T_r}^1(v)\}$. The *level* of a vertex v in T_r is the distance $d_{T_r}(v, r)$, i.e., the number of edges in the path between v and r . For $v \in V$, T_v is the subtree of T_r containing v once the edge between v and its father has been deleted from T_r . Hence, v will be considered as a root of the subtree T_v .

Let us fix arbitrarily the vertex 1 as the root of the tree T_1 ; in Figure 4.3, we have a directed edge from a vertex v to its father. Then $\text{fa}_{T_1}^1(1) = \emptyset$, $\text{fa}_{T_1}^1(5) = 2$ and $\text{fa}_{T_1}^2(5) = 1$. For instance, vertices 5, 6 are leaves. T_2 is the subtree of T_1 rooted at 2, containing vertices $\{2, 4, 5\}$.

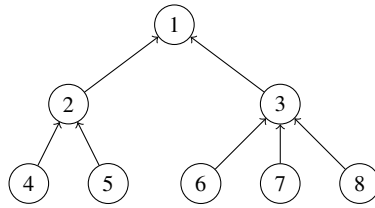


Figure 4.3: Example of a tree T_1 rooted at vertex 1.

We now characterize the yes-instances of EXT VC in forests by a kind of forbidden structure. Consider a tree $T = (V, E)$ and a set of vertices U . A subtree $T' = (V', E')$ of a tree T is called *full with respect to (T, U)* if $U \subseteq V'$, $d_{T'}(u) = d_T(u)$ for all $u \in U$. T' is *minimally full with respect to (T, U)* if it is full with respect to U and for any $e \in E'$, $T' - e$ is not full with respect to U . Finally, a subtree $T' = (V', E')$ is *induced full with respect to (T, U)* if it is full with respect to $(T, U \cap V')$.

Consider the following class of trees \mathcal{T} containing black and white vertices (where the resulting 2-coloring is not proper but each vertex has a color white or black), defined inductively by:

- Base case: A tree with a single vertex x belongs to \mathcal{T} if x is black.
- If $T \in \mathcal{T}$, then the tree resulting from the addition of a P_3 (3 new vertices that form a path p) where one endpoint of p is black, the two other vertices are white and the white endpoint of p is linked to any black vertex of T is in \mathcal{T} .

Example 4.11. There are five black-and-white trees on at most ten vertices in \mathcal{T} : four paths (on one, four, seven and ten vertices), where the endpoints are black and otherwise every third vertex is black, and one is a subdivided star, whose center is black and of degree three, and the three black leaves are at distance three from the center (see Figure 4.4).

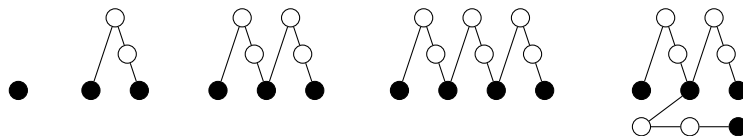


Figure 4.4: Five different black-and-white trees on at most ten vertices in \mathcal{T} .

Clearly, we can define a 2-coloring by describing the set of black vertices. More specifically, if $T = (V, E)$ is a tree and $X \subseteq V$, then $T[X \rightarrow \text{black}]$ denotes the black-and-white tree where exactly the vertices from X are colored black.

Remark 4.12. By induction, it is easy to see that any leaf of any tree in \mathcal{T} is black. Again inductively, one sees that for any black-and-white tree $T \in \mathcal{T}$, all vertices at distance one or two from a black vertex v are white, while all vertices at distance three from v are black.

We are ready now to characterize the solutions in trees.

Theorem 4.13. *Let $T = (V, E)$ be a tree and $U \subseteq V$ be an independent set. Then, (T, U) is a yes-instance of EXT VC if and only if there is no subtree $T' = (V', E')$ of T that is induced full with respect to (T, U) such that $T'[U \rightarrow \text{black}] \in \mathcal{T}$.*

Proof. First, observe that if $U = \emptyset$, then (T, U) is clearly a yes-instance of EXT VC, while subtrees $T'[U \rightarrow \text{black}]$ would contain white vertices only and hence would never belong to \mathcal{T} . Hence, in this case, the assertion of the theorem is clearly satisfied, so that we can assume $U \neq \emptyset$ in the following reasoning.

The condition is necessary. Consider an instance (T, U) of EXT VC such that T contains a subtree $T'[U \rightarrow \text{black}] \in \mathcal{T}$ that is induced full with respect to (T, U) . Then, the leaves of T' are some leaves of T . By Remark 4.12, these vertices belong to U . If (T, U) is a yes-instance of EXT VC, then, according to item (iii) of Theorem 4.3, there exists an independent dominating set $S' \subseteq N_T[U] \setminus U$ of $T[N_T[U]]$. Consider a black leaf u . Its neighbor v necessarily belongs to the independent dominating set S' and then the neighborhood N of v (other than u) does not belong to S' . Then, using inductive definition of \mathcal{T} , one new black vertex u' is a neighbor of N and then by repeating the process one neighbor of u' must be a part of the independent dominating set, and so on. At the end of the process we get a contradiction because we end by a leaf which is black and which is not dominated by S' .

The other direction is proved by induction on n , the number of vertices of the considered tree T . If $n \leq 3$, then an exhaustive search proves that the only black-and-white subtree in \mathcal{T} contains one (black) vertex only (see Example 4.11). This can be a subtree that is induced full with respect to a vertex set that clearly contains this black vertex only (by construction) if and only if $n = 1$. In that case, (T, U) is a no-instance of EXT VC. If T contains two or three vertices, then there is no way to find a black-and-white subtree in \mathcal{T} that is induced full with respect to (T, U) ; observe that U necessarily contains one or two vertices, because U is independent. Yet, it is also clear that (T, U) is a yes-instance to EXT VC if T has two or three vertices and U is independent.

Assume the result is valid for any tree of at most n vertices satisfying the hypothesis of Theorem 4.13 and let (T, U) be an instance of EXT VC, where T is a tree of $n + 1 \geq 4$ vertices, U is an independent set, and T does not contain any subtree T' that is induced full with respect to (T, U) , such that $T'[U \rightarrow \text{black}] \in \mathcal{T}$. As said above, we can assume $U \neq \emptyset$. Moreover, T is not a star $K_{1,p}$ because one of the two minimal vertex cover containing U is a certificate (recall U is supposed to be an independent set).

Set $r = v \in V \setminus U$ with $d_T(v) \geq 2$ be a root of $T = T_r$ using previous notations. There is such vertex since $n \geq 4$, U is an independent set and $T \neq K_{1,p}$. Consider two cases:

- T has no leaves in U . For each $u \in U$, let $v_u \in \text{ch}_{T_r}(u)$ be any child of u . The set $S = \{v_u : u \in U\}$ satisfies item (iii) of Theorem 4.3 and then (T, U) is a yes-instance of EXT VC.
- T admits some leaves in U . Let $u \in U$ be a leaf which has a grandfather in T_r different from r ; if such vertex does not exist, then $S = \{v_u : u \in U\} \cup \{r\}$ satisfies item (iii) of Theorem 4.3, where v_u is defined as in the previous item.

Otherwise, let $v = \text{fa}_{T_r}^2(u)$ be the grandfather of u with $v \neq r$. Consider the two following cases: $v \in U$ and $v \notin U$. If $v \in U$, then let $T_v = T - \{u\}$. By construction, $(T_v, U \setminus \{u\})$ satisfies the hypothesis of Theorem and T_v has strictly less than $n + 1$ vertices. Hence, there is a minimal vertex cover S in T_v which contain $U \setminus \{u\}$ (and then not $\text{fa}_{T_r}(u)$). So, $S \cup \{u\}$ is a minimal vertex cover of T containing U .

Now, assume $v \notin U$ and consider the two subtrees T_v and T_w resulting from the deletion of the edge between v and its father $w = \text{fa}_{T_r}^3(u)$. Let U_v and U_w be the vertices of U inside T_v and T_w , respectively. These two trees have strictly less than $n + 1$ vertices. By construction, (T_v, U_v) satisfies the condition of the theorem because it is induced full with respect to (T, U) . Then by inductive hypothesis, there is a minimal vertex cover S_v of T_v with $U_v \subseteq S_v$. Now, assume that (T_w, U_w) does not satisfy the condition of the theorem.

This means that $w \in U_w$, is a leaf of T_w and there is $T' \in \mathcal{T}$ containing w . Since U is an independent set, $(T' \cup \{\text{ufa}_{T_r}^1(u), \text{fa}_{T_r}^1(u)\text{fa}_{T_r}^2(u), \text{fa}_{T_r}^2(u)\text{fa}_{T_r}^3(u)\}) \in \mathcal{T}$ and is induced full with respect to (T, U) which is a contradiction.

Hence, using the induction hypothesis, there is a minimal vertex cover S_w of T_w with $U_w \subseteq S_w$. In conclusion, $S_v \cup S_w$ is a certificate. \square

Using Theorem 4.13, we are able to produce a linear-time algorithm:

Theorem 4.14. *EXT VC can be solved in linear-time in forests.*

Proof. We proceed as in the last part of Theorem 4.13. First, we delete edges between the required vertices of the given instance in order to obtain an independent set U ; see Remark 4.2. Also, we (arbitrarily) define roots for every tree (connected component) of the forest. For each connected component T with root r of the forest, we find (if any) a leaf in $u \in U$ with largest level. We separate T into T_v and T_1 where $v = \text{fa}_{T_r}^2(u)$. In T_v , we search a path p in \mathcal{T} on four vertices which contains v . If one has been found, we return no and otherwise we apply the same procedure on T_1 , that is, if T_1 is not a leaf in U (corresponding to a base case graph of \mathcal{T}) we find (if any) a leaf in $u \in U$ with largest possible level and so on. \square

In the following, we study the parameterized complexity of EXT VC (resp. EXT IS) with respect to the size of U (resp. $V \setminus U$) for a given graph $G = (V, E)$ and a forced (co-forbidden) set U .

4.2.2 Parameterized complexity

We now study our problems in the framework of parameterized complexity where we consider the size of the set of forced vertices as *standard parameter* for our extension problems. For our hardness proof we give a reduction from MULTICOLORED INDEPENDENT SET. The input of MULTICOLORED INDEPENDENT SET consists of a graph G , an integer k , and a partition (V_1, \dots, V_k) of the vertices of G ; the task is to decide if there is an independent set of size k , containing exactly one vertex from each set V_i .

Theorem 4.15. *EXT VC with standard parameter is $W[1]$ -complete, even when restricted to bipartite instances.*

Proof. We show hardness by reduction from MULTICOLORED INDEPENDENT SET. So let $G = (V, E)$ be an instance of MULTICOLORED INDEPENDENT SET, with partition V_1, \dots, V_k for V . W.l.o.g., assume that each V_i induces a clique and $|V_i| \geq 2$. Construct $G' = (V', E')$ from G with V' built from two copies of V , denoted V and $\bar{V} := \{\bar{v} : v \in V\}$, and $2k$ additional vertices $\{w_i, \bar{w}_i : 1 \leq i \leq k\}$, and E' containing $u\bar{v}$ for all $uv \in E$ and $u\bar{w}_i$ and $\bar{u}w_i$ for all $u \in V_i$, $i \in \{1, \dots, k\}$ (see Figure 4.5). G' is bipartite with partition into $V \cup \{w_i : 1 \leq i \leq k\}$ and $\bar{V} \cup \{\bar{w}_i : 1 \leq i \leq k\}$. Set $U = \{w_i, \bar{w}_i : 1 \leq i \leq k\}$ and consider (G', U) as instance of EXT VC. We claim that (G', U) is a *yes*-instance for EXT VC if and only if G is a *yes*-instance for MULTICOLORED INDEPENDENT SET. Since MULTICOLORED INDEPENDENT SET is $W[1]$ -hard [79],² this FPT-reduction shows $W[1]$ -hardness for EXT VC with standard parameterization.

Suppose (G', U) is a *yes*-instance for EXT VC, so there exists a minimal vertex cover S for G' with $U \subseteq S$. Consider $S' := V' \setminus S$. Since S is minimal, $N(u) \not\subseteq S$ for all $u \in S$, so especially for each $i \in \{1, \dots, k\}$ there exists at least one vertex from $N(w_i) = \bar{V}_i$ in S' and also at least one vertex from $N(\bar{w}_i) = V_i$ in S' . Since S' has to be an independent set in G' and $v\bar{u} \in E'$ for all $u, v \in V_i$, $u \neq v$ (recall that V_i is a clique in G), it follows that if $v \in S' \cap V_i$, then \bar{v} is the only vertex independent from v in \bar{V}_i . This means that $|S' \cap V_i| = 1$ for all $i \in \{1, \dots, k\}$ and if $S' \cap V = \{v_1, \dots, v_k\}$, then $S' \cap \bar{V} = \{\bar{v}_1, \dots, \bar{v}_k\}$. The set $S' \cap V$ hence is a multicolored independent set in G , since $v_i v_j \in E$ for $i, j \in \{1, \dots, k\}$ would imply that $v_i \bar{v}_j \in E'$ which is not possible since S' is an independent set in G' . Conversely, it is not hard to see that if there exists a multicolored independent set S in G , then the set $V' \setminus (S \cup \bar{S})$ (with $\bar{S} := \{\bar{v} : v \in S\}$) is a minimal vertex cover for G' containing U .

Membership in $W[1]$ is seen as follows. As suggested in [42], we describe a reduction to SHORT NONDETERMINISTIC TURING MACHINE problem which is already explained in Chapter 1. Given a graph $G = (V, E)$ and a pre-solution $U = \{u_1, \dots, u_k\} \subseteq V$, the constructed Turing machine first guesses vertices u'_1, \dots, u'_k , with $u'_i \in N(u_i) \setminus U$ and then verifies in time $O(k^2)$ if the guessed set U' is an independent set. As $\{u'_1, \dots, u'_k\}$ can be greedily extended to an independent dominating set for $N[U]$ which, by Theorem 4.3, is equivalent to (G, U) being a *yes*-instance of EXT VC, U can be extended to a minimal vertex cover if and only if one of the guesses is successful. \square

Considering condition (ii) in Theorem 4.3, it is obvious to see that for an instance $(G = (V, E), U)$, EXT IS with respect to the dual parameter $|(V \setminus U)|$ is $W[1]$ -complete.

Corollary 4.16. *EXT IS with respect to the dual parameter is $W[1]$ -complete, even when restricted to bipartite instances.*

As a remark, it is obvious to see that considering the parameter $|U|$ instead of $|V \setminus U|$ leads to an FPT-result for EXT IS, as it is sufficient to test if any of the subsets of $V \setminus U$, forms a maximal independent set. However, this algorithm is quite trivial and we can do it in time $O^*(2^{|U|})$.

Remark 4.17. EXT IS with standard parameter is in FPT.

Considering condition (ii) in Theorem 4.3, we easily deduce:

²The proof is for MULTICOLORED CLIQUE; taking the complement graph is a parameterized reduction showing that MULTICOLORED INDEPENDENT SET is $W[1]$ -hard.

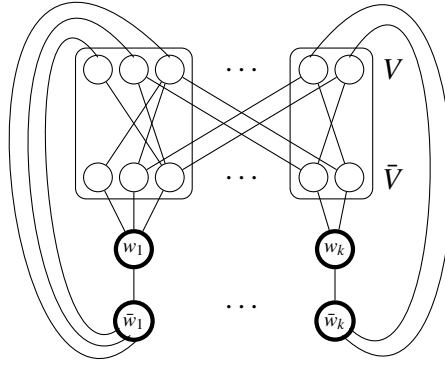


Figure 4.5: The graph $G' = (V', E')$ for EXT VC. Vertices in U are drawn bold.

Remark 4.18. EXT VC with dual parameter is in FPT.

By distinguishing between whether a vertex belongs to the cover or not and further, when it belongs to the cover, if it already has a private edge or not, it is not hard to design a dynamic programming algorithm that decides in time $O^*(c^t)$ if (G, U) is a yes-instance of EXT VC or not, given a graph G together with a tree decomposition of width t . Similar to what we did in Chapter 3 in the dynamic programming algorithm, we consider two different labels *cover* and *independent* for any vertex in the tree decomposition, and try to keep all possible labelings which leads to a minimal vertex cover by considering that all the vertices in U are already labeled with *cover*. With some more care, even $c = 2$ can be achieved, but this is not so important here. Moreover, it is well known that planar graphs of order n have treewidth bounded by $O(\sqrt{n})$ [98]. In fact, we can obtain a corresponding tree decomposition in polynomial time, given a planar graph G . Piecing things together, we obtain:

Theorem 4.19. EXT VC can be solved in time $O^*(2^{O(\sqrt{n})})$ on planar graphs of order n .

Let (G, U) be an instance of EXT VC such that G is planar. By Theorem 4.3, it suffices to solve EXT VC on (G', U) , where G' is the graph induced by $N_G[U]$. Clearly, G' is also planar. Moreover, the diameter of each connected component of G' is upper-bounded by $3|U|$, observe that we only consider for the graph G' the neighborhood of the independent set U . Therefore, G' is (at most) $3|U|$ -outerplanar and hence according to [27], the treewidth of G' is at most $9|U|$. Our previous remarks show that EXT VC can be solved in time $O^*(2^{O(|U|)})$.

Theorem 4.20. EXT VC (resp. EXT IS) with standard parameter (resp. dual parameter) is in FPT on planar graphs.

4.2.3 Price of extension

From an instance $I = (G, U)$ of EXT VC and EXT IS, we define two new optimization problems respectively as follows:

$\text{EXT}_{\max}\text{VC}$

Input: A graph $G = (V, E)$, a set of vertices $U \subseteq V$.

Solutions: Minimal vertex cover S of G .

Output: Solution S that maximizes $|S \cap U|$.

$\text{EXT}_{\min}\text{IS}$

Input: A graph $G = (V, E)$, a set of vertices $U \subseteq V$.

Solutions: Maximal independent set S of G .

Output: Solution S that minimizes $|S \cup U|$.

For $\Pi = \text{EXT}_{\max}\text{VC}$ or $\text{EXT}_{\min}\text{IS}$, we denote by $\text{opt}_{\Pi}(G, U)$ the value of an optimal solution. Notice that alternatively these two optimal quantities can be expressed as

- $\text{opt}_{\text{EXT}_{\max}\text{VC}}(G, U) = \arg \max\{U' \subseteq U : (G, U') \text{ is a yes-instance of EXT VC}\}$,
- $\text{opt}_{\text{EXT}_{\min}\text{IS}}(G, U) = \arg \min\{U' \supseteq U : (G, U') \text{ is a yes-instance of EXT IS}\}$.

Since for both of them $\text{opt}_{\Pi}(G, U) = |U|$ if and only if (G, U) is a *yes*-instance of the extension variant, we deduce that $\text{EXT}_{\max}\text{VC}$ and $\text{EXT}_{\min}\text{IS}$ are NP-hard since EXT VC and EXT IS are NP-complete. Similarly to Remark 4.1, one observes that the decision variants of $\text{EXT}_{\max}\text{VC}$ and $\text{EXT}_{\min}\text{IS}$ are indeed completely equivalent. More precisely,

$$\text{opt}_{\text{EXT}_{\max}\text{VC}}(G, U) + \text{opt}_{\text{EXT}_{\min}\text{IS}}(G, U') = |V| \quad (4.1)$$

Adapted by Definition 2.4, we define the concept of *price of extension* for two $\text{EXT}_{\max}\text{VC}$ and $\text{EXT}_{\min}\text{IS}$ as follows:

Definition 4.21. For an instance (G, U) of $\text{EXT}_{\max}\text{VC}$, $\text{PoE}(S)$ of a minimal vertex cover S of G is defined by $\text{PoE}(S) := \frac{|S \cap U|}{\text{opt}_{\text{EXT}_{\max}\text{VC}}(G, U)}$. Similarly, $\text{PoE}(S) := \frac{|S \cup U|}{\text{opt}_{\text{EXT}_{\min}\text{IS}}(G, U)}$ is the price of extension of a maximal independent set S for an instance (G, U) of $\text{EXT}_{\min}\text{IS}$.

Negative results on the price of extension

Now, we propose negative bounds of the price of extension even in special cases. Dealing with $\text{EXT}_{\max}\text{VC}$ on graph $G = (V, E)$ and the particular subset $U = V$ (resp. $\text{EXT}_{\min}\text{IS}$ with $U = \emptyset$), we obtain two well known optimization problems called *Upper Vertex Cover* or *maximum minimal vertex cover*, denoted by MAX-MIN VC and *Minimum Independent Dominating Set* or *minimum maximal independent set*, denoted by MIN-MAX IS. In [137], the computational complexity of these problems are studied (among 12 problems), and (in)approximability results are given in [141, 30] for MAX-MIN VC and in [100] for MIN-MAX IS where lower bounds of $O(n^{\varepsilon-1/2})$ and $O(n^{\varepsilon-1})$, respectively, for graphs on n vertices are given for every $\varepsilon > 0$. Analogous bounds can be derived depending on the maximum degree Δ . In particular, we deduce:

Corollary 4.22. For any constant $\varepsilon > 0$, any $\rho \in \Omega(n^{\varepsilon-1})$ and $\rho \in \Omega(\Delta^{\varepsilon-1})$, there is no polynomial-time ρ -approximation for $\text{EXT}_{\min}\text{IS}$ on general graphs of n vertices and maximum degree Δ , even when $U = \emptyset$, unless $P = NP$.

Now, we strengthen the lower bounds of $\Omega(n^{\varepsilon-1/2})$ and $\Omega(\Delta^{\varepsilon-1/2})$ for $\text{Ext}_{\max}\text{VC}$.

Theorem 4.23. *$\text{Ext}_{\max}\text{VC}$ is as hard as MAX IS to approximate in general graphs even if the set U of forced vertices forms an independent set.*

Proof. The proof is based on a simple reduction from MAX IS . Let $G = (V, E)$ be an instance of MAX IS , construct the graph $H = (V_H, E_H)$ from G , where vertex set V_H contains two copies of V , denoted by V and $V' = \{v' : v \in V\}$. The edge set E_H contains E together with vv' for all $v \in V$, formally, $E_H = E \cup \{vv' : v \in V\}$. Consider $I = (H, U)$ as instance of $\text{Ext}_{\max}\text{VC}$, where the forced vertex subset is given by $U = V'$. We claim, H has a minimal vertex cover containing k vertices from U if and only if G has a maximal independent set of size k .

Let S be a maximal independent set of G of size k ; then $S' = \{v : v \notin S\} \cup \{v' : v \in S\}$ is a minimal vertex cover of H containing k vertices from U . Conversely, let S' be a minimal vertex cover of H extending U , with $U' = U \cap S'$. By construction, the set $S' \setminus U'$ is a vertex cover of G and then $S = V \setminus S'$ is an independent set of G of size $|U'|$. In particular, we deduce $\alpha(G) = \text{opt}_{\text{Ext}_{\max}\text{VC}}(H, U)$. \square

Using the strong inapproximability results for MAX IS given in [154, 165], observing $\Delta(H) = \Delta(G) + 1$ and $|V_H| = 2|V|$, we deduce the following result.

Corollary 4.24. *For any constant $\varepsilon > 0$, any $\rho \in \Omega(\Delta^{1-\varepsilon})$ and $\rho \in \Omega(n^{1-\varepsilon})$, there is no polynomial-time ρ -approximation for $\text{Ext}_{\max}\text{VC}$ on general graphs of n vertices and maximum degree Δ , unless $P = NP$.*

Positive results on the price of extension

In contrast to the hardness results on these restricted graph classes from the previous sections, we find that restriction to bipartite graphs or graphs of bounded degree improve approximability of $\text{Ext}_{\max}\text{VC}$. For the following results, we assume, w.l.o.g., that the input graph is connected, non-trivial and therefore without isolated vertices, as we can solve our problems separately on each connected component and then combine the results.

Theorem 4.25. *A 2-approximation for $\text{Ext}_{\max}\text{VC}$ on bipartite graphs can be computed in polynomial time.*

Proof. Let $G = ((V = V_l \cup V_r), E)$ and $U \subseteq V$ be an instance of $\text{Ext}_{\max}\text{VC}$, where E contains only edges connecting V_l and V_r . Since V_l and V_r are both minimal vertex covers (G is without isolated vertices) and also a partition of V , then taking one of them containing the largest number of vertices from U (assume it is V_l), we get a 2-approximation, because $2 \times |V_l \cap U| \geq |V \cap U| \geq \text{opt}_{\text{Ext}_{\max}\text{VC}}(G, U)$. \square

Theorem 4.26. *A Δ -approximation for $\text{Ext}_{\max}\text{VC}$ on graphs of maximum degree Δ can be computed in polynomial time.*

Proof. Let $G = (V, E)$ be connected of maximum degree Δ , and $U \subseteq V$ be an instance of $\text{EXT}_{\max} \text{VC}$. If $\Delta \leq 2$, or if $G = K_{\Delta+1}$ (the complete graph on $\Delta + 1$ vertices), it is easy to check $\text{EXT}_{\max} \text{VC}$ is polynomial-time solvable; actually in these two cases, G is either chordal or circular-arc and Theorem 4.27 gives the conclusion. Hence, assume $\Delta \geq 3$ and $G \neq K_{\Delta+1}$. By Brooks's Theorem, we can color G properly with at most Δ colors in polynomial-time (even linear). Let (S_1, \dots, S_ℓ) be such coloring of G with $\ell \leq \Delta$. For $i \leq \ell$, set $U_i = U \cap N_G(S_i)$ where we recall $N_G(S_i)$ is the open neighborhood of S_i . By construction, S_i is an independent set which dominates U_i in G so it can be extended to satisfy (iii) of Theorem 4.3, so (G, U_i) is a *yes*-instance of EXT VC . Choosing $U' = \arg \max |U_i|$ yields a Δ -approximation, since on the one hand $\sum_{i=1}^{\ell} |U_i| \geq |U \cap (\cup_{i=1}^{\ell} N_G(S_i))| = |U \cap V|$ and on the other hand $\Delta \times |U'| \geq \sum_{i=1}^{\ell} |U_i| \geq |U| \geq \text{opt}_{\text{EXT}_{\max} \text{VC}}(G, U)$. \square

Along the lines of Corollary 4.10 with more careful arguments, we can prove:

Theorem 4.27. *$\text{EXT}_{\max} \text{VC}$ can be solved optimally for chordal graphs and circular-arc graphs in polynomial time.*

Proof. Let (G, U) be an instance of $\text{EXT}_{\max} \text{VC}$ where $G = (V, E)$ is a chordal graph (resp. a circular-arc graph) and U is an independent set. We build a weighted graph G' for WMIN ISDS such that G' is the subgraph of G induced by $N_G[U]$ and the weights on vertices are given by $w(v) = 1$ if $v \in U$ and $w(v) = 0$ for $v \in N_G[U] \setminus U$. Thus, we get: $\text{opt}_{\text{WMIN ISDS}}(G', w) = |U| - \text{opt}_{\text{EXT}_{\max} \text{VC}}(G, U)$. \square

4.2.4 Generalizations to extensions of H -graph cover and H -free subgraph

Assume that graph $H = (V_H, E_H)$ is fixed; the $\text{MAXIMUM INDUCED } H\text{-FREE SUBGRAPH PROBLEM}$, $\text{INDUCED } H\text{-FREE}$ for short, asks, given a graph $G = (V, E)$, to find the largest subset of vertices $S \subseteq V$ such that the subgraph $G[S]$ induced by S is H -free, i.e. it does not contain any copy of H . A corresponding extension version is given by:

$\text{EXT INDUCED } H\text{-FREE}$

Input: A graph $G = (V, E)$, a set of vertices $U \subseteq V$.

Question: Does G have a maximal subgraph $G[S]$ induced by S with $S \subseteq U$ which is H -free?

The particular case of $H = K_2$ corresponds to EXT IS , because S induces a K_2 -free subgraph if and only if it is an independent set. We now generalize our previous results on the complexity of EXT IS towards this more general setting. Recall that a graph is *biconnected* if it stays connected after deleting any single vertex.

Theorem 4.28. *If H is biconnected with at least 2 vertices, then $\text{EXT INDUCED } H\text{-FREE}$ is NP-complete.*

Proof. Let $H = (V_H, E_H)$ with $n_H = |V_H|$ vertices be a biconnected graph and assume $n_H \geq 3$ ($n_H = 2$ corresponds to EXT IS which has been proved NP-complete in Theorem 4.4). The

proof is based on a reduction from H -FREE 2-COLORABILITY, denoted by H -2COL for short. With fixed H , the problem H -2COL consists in deciding if the vertices of a given graph $G = (V, E)$ can be partitioned into two H -free induced subgraphs G_{V_i} , $i = 1, 2$ (so $V = V_1 \cup V_2$). H -2COL is NP-complete if and only if H contains at least 3 vertices; see [3]. From $G = (V, E)$ with n vertices, as an instance of H -2COL, we build an instance of EXT INDUCED H -FREE as follows:

Let u and w be two distinct vertices of H . We consider two copies G_1 and G_2 of G where v^1 and v^2 are copies of vertex $v \in V$ and $2n$ copies H_i of H (where u_i and w_i are copies of u and w). We collapse together two copies H_i and H_{n+i} by merging vertices w_i and w_{n+i} ; Let H'_i be the resulting graph and w'_i be the vertex corresponding to w_i and w_{n+i} after the merging. Now, we merge vertices v_i^1 with u_i and v_i^2 with u_{n+i} and we get the graph G' as part of an instance of EXT INDUCED H -FREE. Hence, G' contains G_1 , G_2 and the graphs H'_i for $i = 1, \dots, n$. Finally, we set $U = V(G') \setminus \{w'_i : 1 \leq i \leq n\}$.

Now, we claim that G is a *yes*-instance of H -2COL if and only if (G', U) is a *yes*-instance of EXT INDUCED H -FREE. Assume that (V_1, V_2) is an H -free 2-coloring (bipartition) of G . Consider any maximal H -free subgraph of G containing V_1 (resp. V_2) and call it V'_1 (resp. V'_2). Finally, let us denote by $V(H'_i)$ the set of vertices of subgraph H'_i for each $i \in \{1, \dots, n\}$. We claim that the set $S = \bigcup_{i=1}^n (V(H'_i) \setminus \{w'_i, v_i^1, v_i^2\}) \cup \bigcup_{j=1}^2 \{v_i^j : v_i \in V'_j\}$ induces a maximal H -free subgraph of G' with $S \subseteq U$. Actually, this is clear inside either each copy of G or each $V(H'_i) \setminus \{w'_i\}$. An assumed copy R of H in S must hence include vertices in a copy G_j with $j = 1, 2$ and also vertices in a copy H'_i for some $i = 1, \dots, n$. Hence, to be connected, R has to contain some v_i^j which is a cut-vertex of R , separating the vertices in R from $V(G_j) \setminus \{v_i^j\}$, for some $j \in \{1, 2\}$, from $V(H'_i) \setminus \{v_i^j\}$, which is a contradiction to H being biconnected.

Conversely, assume that there exists a set $S \subseteq U$ which induces a maximal H -free subgraph of G' . Let S_i for $i = 1, 2$ be the vertices of S included in copy G_i . Let $V_1 = \{v_j : v_j^1 \in S_1\}$ and $V_2 = \{v_j : v_j^2 \in S_2 \text{ and } v_j^1 \notin S_1\}$. We claim that (V_1, V_2) is an H -free 2-coloring of G . Obviously, each subgraph of G induced by V_i is H -free for $i = 1, 2$. If (V_1, V_2) is not a vertex partition of G , then there exists some $i \in \{1, \dots, n\}$ such that $v_i^j \notin S$ for both $j = 1$ and $j = 2$ for some $i = 1, \dots, n$. This however implies that $(V(H'_i) \setminus \{v_i^1, v_i^2\}) \subseteq S$ because $(V(H'_i) \setminus \{v_i^1, v_i^2\})$ is H -free (recall that H is assumed to be biconnected). Thus, $w'_i \in S$ which is a contradiction to $S \subseteq U$. \square

We are now considering a covering analogue to EXT INDUCED H -FREE.

EXT H -COVER

Input: A graph $G = (V, E)$, a set of vertices $U \subseteq V$.

Question: Does G have a minimal subset S which covers all copies of H with $U \subseteq S$?

Similarly to previous remark, we have EXT K_2 -COVER = EXT VC. More generally for any fixed graph H , S is a minimal H -cover of $G = (V, E)$ if and only if $V \setminus S$ is a maximal H -free subgraph. Hence, using same the reasoning that the one given in Remark 4.1, we deduce:

Proposition 4.29. *If H is biconnected, then EXT H -COVER is NP-complete.*

Notice that this last assertion is interesting, as for the corresponding classical problem H -COVER, no easy conditions like biconnectivity are known to yield NP-completeness results, see [1, 128, 129]. We are now stating a characterization of graphs admitting an H -cover extension that could be compared to Theorem 4.3.

Theorem 4.30. *Let $G = (V, E)$ be a graph and $U \subseteq V$ be a set of vertices. There is a minimal H -cover S of G extending U if and only if the two following conditions hold:*

- (i) *For every $u \in U$, there is a copy $H_u = (V(H_u), E_u)$ of H in G such that $V(H_u) \cap U = \{u\}$.*
- (ii) *If $V' = \bigcup_{u \in U} V(H_u)$, then the subgraph G' of G induced by $V' \setminus U$ is H -free.*

Proof. Let $G = (V, E)$ be a graph and a set $U \subseteq V$ be a set of vertices. The condition is sufficient; indeed, let S' be any maximal H -free subgraph containing V' where V' is defined according to conditions (i) and (ii) (via H_u). The set $V \setminus S'$ is a minimal H -cover of G extending U .

Conversely, assume that S is a minimal H -cover of G extending U . Let us prove that S satisfies conditions (i) and (ii). Since $U \subseteq S$ is a minimal H -cover of G , then for every $u \in U$, there exists a copy $H_u = (V(H_u), E_u)$ of H in G such that $u \in V(H_u)$ and $S \setminus \{u\}$ does not cover H_u . In particular, we deduce $V(H_u) \cap U = \{u\}$. Now, let $V' = \bigcup_{u \in U} V(H_u)$; if the subgraph G' of G induced by $V' \setminus U$ is not H -free, then $\exists v \in (V' \cap S) \setminus U$, such that v lies in $V(H_{u_0})$ for some $u_0 \in U$, contradicting the fact that $S \setminus \{u_0\}$ does not cover H_{u_0} . \square

Corollary 4.31. *For every fixed H , EXT H -COVER parameterized by $|U|$ is in XP.*

Proof. Using exhaustive search, finding $V' = \bigcup_{u \in U} V(H_u)$ can be done in time $O(n^{k+n_H})$, where $k = |U|$ and $n_H = |V(H)|$. The remaining steps can be performed in $O(n^{k+n_H}) = O(n^k)$ time, as H is fixed. \square

4.3 EXT CVC and EXT NSIS

In this section, we consider the extension variants of two optimization problems CONNECTED VERTEX COVER problem and NON SEPARATING INDEPENDENT SET problem. A subset $S \subseteq V$ of a connected graph $G = (V, E)$ is a *connected vertex cover* if S is a *vertex cover* (i.e. each edge of G is incident to at least a vertex of S) and the subgraph $G[S]$ induced by S is connected. The corresponding optimization problem MINIMUM CONNECTED VERTEX COVER (MIN CVC for short) consists in finding a connected vertex cover of minimum size. A non separating independent set S of a connected graph $G = (V, E)$ is a subset of vertices of G which is *independent* (i.e. any two vertices in S are non adjacent) and S is not a cut-set of G . The corresponding optimization problem MAXIMUM NON SEPARATING INDEPENDENT SET (MAX NSIS for short) asks to find a non separating independent set of maximum size. MIN CVC and MAX NSIS have been studied in [87, 74, 156, 83, 148] where it is proved that the problems are polynomially solvable in graphs of maximum degree 3, while in graphs of maximum degree 4 they are NP-hard. Here, we study the extension variant of these two problems: *extension connected vertex cover* denoted by EXT CVC

consists of a connected graph $G = (V, E)$ together with a subset $U \subseteq V$ of vertices as an input, and the goal is to decide whether there exists a minimal connected vertex cover of G containing U . A subset $S \subseteq V$ is a minimal connected vertex cover if and only if for any $S' \subset S$, either S' is not a vertex cover or it is not connected. *Extension non separating independent set* denoted by EXT NSIS consists of a connected graph $G = (V, E)$ together with a subset $U \subseteq V$ of vertices as an input, and the goal is to decide whether there exists a maximal non separating independent set of G excluding vertices from $V \setminus U$. The formal definitions of two problems are as follows:

EXT CVC

Input: A connected graph $G = (V, E)$ and a subset $U \subseteq V$.

Question: Does G have a minimal connected vertex cover S with $U \subseteq S$?

EXT-NSIS

Input: A connected graph $G = (V, E)$ and a subset $U \subseteq V$.

Question: Does G have a maximal non separating independent set S with $S \subseteq U$?

Recall that, in EXT CVC (resp. EXT NSIS), the set U is also referred to as the set of *forced* (resp. *co-forbidden*). In the following, we will study the two problems in the framework of computational complexity.

4.3.1 Computational complexity

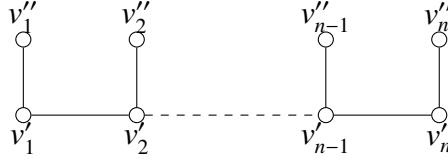
Let us begin by some simple observations: (G, U) with $G = (V, E)$ and $U \subseteq V$ is a *yes*-instance of EXT CVC if and only if $(G, V \setminus U)$ is a *yes*-instance of EXT NSIS. Hence, all complexity results given in this section for EXT CVC are valid for EXT NSIS as well. A leaf ($v \in V_l$) never belongs to a minimal connected vertex cover S (apart from the extreme case where G consists of a single edge), while any cut-vertex $v \in V_c$ necessarily belongs to S . This implies that for trees, we have a simple characterization of *yes*-instances for $n \geq 3$: (T, U) , where $T = (V, E)$ is a tree, is a *yes*-instance of EXT CVC if and only if U is a subset of cut-set V_c , or equivalently $U \subseteq V_c = V \setminus V_l$. For an edge or a cycle $C_n = (V, E)$, (C_n, U) is a *yes*-instance if and only if $U \neq V$; for a path $P_n = (V, E)$, (P_n, U) is a *yes*-instance if and only if $U \subseteq V \setminus V_l$, which settles the case of graphs of maximum degree 2. Dealing with split graphs, a similar but more complicated characterization can be given. In the following, we will deduce more general results for EXT CVC by showing and exploiting relations to EXT VC.

Relation between EXT VC and EXT CVC

The following two properties allow to make use of known results for EXT VC to obtain results for EXT CVC.

Proposition 4.32. *EXT CVC is polynomially reducible to EXT VC.*

Proof. We propose a simple reduction from EXT CVC to EXT VC in chordal graphs. Let (G, U) be an instance of EXT CVC where $G = (V, E)$ is a chordal graph and $U \subseteq V$. From G we build an


 Figure 4.6: The additional component H .

instance (G', U') of EXT VC where $G' = (V', E')$ contains V by adding to G a new vertex v' and a pendant edge vv' for each vertex $v \in V_c$ (recall that V_c is the set of cut-vertices of G); it is easy to see G' is also chordal with $|V'| = |V| + |V_c|$. Finally, we set $U' = U \cup V_c$. We claim (G, U) is a *yes*-instance of EXT CVC if and only if (G', U') is a *yes*-instance of EXT VC.

We mainly prove S is a minimal connected vertex cover of G with $U \subseteq S$ if and only if S is a minimal vertex cover of G' containing U' . If S is a minimal connected vertex cover of G , then obviously S is a vertex cover of G and $U' \subseteq S$ because V_c is the set of cut-vertex. If S is not a minimal vertex cover in G' , then $S \setminus \{v\}$ remains a vertex cover of G' for some $v \in S$. Clearly, $v \notin V_c$ because v' needs to be covered and hence this implies by minimality $S \setminus \{v\}$ is disconnected in G which is impossible. Actually, if $S \setminus \{v\}$ is disconnected, then there is a minimal cut-set X of G with $X \cap S = \{v\}$ and $|X| \geq 2$ (because $v \notin V_c$ is not a cut-vertex). Now, X is a clique because G is chordal and then v has a private edge which is a contradiction.

Conversely, if S is a minimal vertex cover of G' containing U' , then $v' \notin S$ for every $v \in V_c$. Thus, by contradiction if S is not connected in G , then there exists a minimal cut-set X of G with $X \cap S = \emptyset$ which separate two distinct components of S . In particular, we must have $|X| \geq 2$ since X does not contain any cut-vertex (which are in S). The set X is a clique outside S (recall G is chordal) and then an edge of G' is not covered by S which is impossible. \square

Proposition 4.33. *EXT CVC is NP-complete in graphs of maximum degree $\Delta + 1$ if EXT VC is NP-complete in graphs of maximum degree Δ , even for bipartite graphs.*

Proof. Given an instance (G, U) of EXT VC, where $G = (V, E)$ with $V = \{v_1, \dots, v_n\}$ and $U \subseteq V$, we build an instance $(G' = (V', E'), U')$ of EXT CVC by adding a component $H = (V_H, E_H)$ to the original graph G .

The construction of H is depicted Figure 4.6 where $V_H = \{v'_i, v''_i : 1 \leq i \leq n\}$ is the vertex set. The new instance of EXT CVC is given by (G', U') and consists of connecting the component H to G by linking $v_i v'_i$ for each $1 \leq i \leq n$ and by setting $U' = U$.

Clearly G' is of maximum degree $\Delta + 1$ if G is of maximum degree Δ . Moreover, it is not difficult to see that (G, U) is a *yes*-instance of EXT VC if and only if (G', U') is a *yes*-instance of EXT CVC.

To maintain bipartiteness, we apply an appropriate subdivision of H . Assume $V = L \cup R$ and $R = \{r_1, \dots, r_p\}$ and $L = \{l_1, \dots, l_q\}$ with $p \leq q$; then, we get $4q$ vertices in V_H where the main path is $\{v'_i : 1 \leq i \leq 2q\}$ and we alternate in this path vertices from L and R and we link r_i to v'_{2i} for $i \leq p$ and l_j to v'_{2j-1} for $j \leq q$. \square

Using polynomial time decidability of EXT VC in chordal graphs, parameterized complexity results (considering that the reduction increases the size of the instances only linearly), and NP-completeness in cubic bipartite graphs [40], we deduce:

Corollary 4.34. *EXT CVC is polynomial-time decidable in chordal graphs and NP-complete in bipartite graphs of maximum degree 4.*

Corollary 4.35. *EXT CVC parameterized with $|U|$ is W[1]-complete, and there is no $2^{o(n+m)}$ -algorithm for n -vertex, m -edge bipartite graphs of maximum degree 4, unless ETH fails.*

Additional hardness results

We first strengthen the hardness result of Corollary 4.34 to bipartite graphs of maximum degree 3. This result may be surprising since the optimization problem MIN CVC is polynomial-time solvable in graphs of maximum degree 3.

Theorem 4.36. *EXT CVC is NP-complete in bipartite graphs of maximum degree 3 even if U is an independent set.*

Proof. We make a reduction from (3, B2)-SAT to EXT CVC. Consider an instance (3, B2)-SAT which clauses $\mathcal{C} = \{c_1, \dots, c_m\}$ and variables $\mathcal{X} = \{x_1, \dots, x_n\}$. We build a bipartite graph $G = (V, E)$ together with a set of forced vertices U as follows:

- For each clause $c = \ell_1 \vee \ell_2 \vee \ell_3$ where ℓ_1, ℓ_2, ℓ_3 are literals, introduce a subgraph $H(c) = (V_c, E_c)$ with 6 vertices and 6 edges. V_c contains three specified literal vertices $\ell_c^1, \ell_c^2, \ell_c^3$. The set of forced vertices in $H(c)$, denoted by U_c is given by $U_c = \{\ell_c^1, \ell_c^2, \ell_c^3\}$. The gadget $H(c)$ is illustrated in the left part of Figure 4.7.
- For each variable x introduce 21 new vertices which induce the subgraph $H(x) = (V_x, E_x)$ illustrated in Figure 4.7. The vertex set V_x contains four special vertices $t_x^{c_1}, t_x^{c_2}, f_x^{c_3}$ and $f_x^{c_4}$, where it is implicitly assumed (w.l.o.g.) that variable x appears positively in clauses c_1, c_2 and negatively in clauses c_3, c_4 . The independent set $U_x = \{1_x, 3_x, 5_x, 6_x, 8_x, 10_x, 12_x\}$ is in U (i.e., forced to be in each feasible solution). The subgraph $H_x - U_x$ induced by $V_x \setminus U_x$ consists of an induced matching of size 5 and of 4 isolated vertices.
- We connect each gadget $H(x_i)$ to $H(x_{i+1})$ by linking vertex 12_{x_i} to vertex $6_{x_{i+1}}$ using an intermediate vertex $r_{i,i+1}$ for all $1 \leq i \leq n - 1$. We also add a pendant edge incident to each $r_{i,i+1}$ with leaf $r'_{i,i+1}$; an illustration of this connection is depicted on the right of Figure 4.7.
- We interconnect $H(x)$ and $H(c)$ where x is a variable occurring in literal ℓ_i of clause c by adding edge $\ell_c^i t_x^c$ (resp., $\ell_c^i f_x^c$), where t_x^c (resp., f_x^c) is in $H(x)$ and ℓ_c^i is in $H(c)$, if x appears positively (resp., negatively) in clause c . These edges are called *crossing edges*.

Let $U = (\bigcup_{c \in C} U_c) \cup (\bigcup_{x \in X} U_x)$. This construction takes polynomial time and G is a bipartite graph of maximum degree 3. Now, we claim that $I = (C, X)$ is satisfiable if and only if G admits a minimal connected vertex cover containing U .

Suppose that I is satisfiable and let T be a truth assignment of I satisfying all clauses. Let us introduce some notation: from T , we consider a minimal *truth literal assignment* L which assigns to each variable x a set of at least one literal and at most two literals according to T such that each clause remains satisfied by L . More formally, for any variable x , $\emptyset \neq L(x) \subseteq \{t_x^{c_1}, t_x^{c_2}\}$ if $T(x) = \mathbf{true}$ and $\emptyset \neq L(x) \subseteq \{f_x^{c_3}, f_x^{c_4}\}$ if $T(x) = \mathbf{false}$. Minimality means here when $L(x) = \{a, b\}$ for some variable $x \in X$, then $L(x) \setminus \{a\}$ or $L(x) \setminus \{b\}$ does not permit to satisfy all clauses (i.e. there is a clause only satisfied by a or b when we consider truth literal assignment L). From L and for each clause c_j , we denote by $c_L(j)$ the set of indices of variables from X satisfying clause c_j in the truth literal assignment L and $nc_L(j) = |c_L(j)|$ its size. By construction, $1 \leq nc_L(j) \leq 3$. Finally, $c_L(j) = c_t(j) \cup c_f(j)$ with $i \in c_t(j)$ (resp. $i \in c_f(j)$) if and only if $L(x_i) \subseteq \{t_x^{c_1}, t_x^{c_2}\}$ (resp. $L(x_i) \subseteq \{f_x^{c_3}, f_x^{c_4}\}$).

We build a connected vertex cover S containing U as follows. First, S contains vertices $\{i_x : 1 \leq i \leq 12\}$ for every variable gadget $H(x)$, vertex set $\{r_{i,i+1} : i = 1, \dots, n\}$ and subset U_c for every clause c . At this stage, we get a minimal vertex cover of G . The new added vertices ensure connectivity of the solution.

- 1– for each variable x_i , add vertices in $L(x_i)$.
- 2– for each clause c_j , depending on the value of $nc_L(j)$, we add exactly $3 - nc_L(j)$ vertices among $\{1_{c_j}, 2_{c_j}, 3_{c_j}\}$ to S as follows:
 - if $nc_L(j) = 3$, then we add nothing,
 - if $nc_L(j) = 2$, then assume that literal $\ell_{c_j}^i$ does not satisfy clause c_j with $i \in \{1, 2, 3\}$. Now, we add to S exactly one arbitrary vertex among $\{1_{c_j}, 2_{c_j}, 3_{c_j}\} \setminus \{i_{c_j}\}$,
 - if $nc_L(j) = 1$, then we add two vertices of $\{1_{c_j}, 2_{c_j}, 3_{c_j}\}$ arbitrarily to S .

It is not difficult to see that S is a minimal connected vertex cover of G containing U .

Conversely, let S be a minimal connected vertex cover of G containing U . By setting, $T(x) = \mathbf{false}$ if $S \cap \{f_x^{c_3}, f_x^{c_4}\} \neq \emptyset$ and $T(x) = \mathbf{true}$ if $S \cap \{t_x^{c_1}, t_x^{c_2}\} \neq \emptyset$, we get a truth assignment satisfying all clauses. Actually, it is not possible to get simultaneously $T(x) = \mathbf{false}$ and $T(x) = \mathbf{true}$ since otherwise $S \setminus \{3_x\}$ shall remain a connected vertex cover of G . \square

Now, we will prove that the polynomial-time decidability of EXT CVC in chordal graphs given in Corollary 4.34 cannot be extended to the slightly larger class of *weakly chordal* (also called *weakly triangulated*³) graphs which are contained in the class of *4-chordal* graphs. For any integer $k \geq 3$, a graph is called *k-chordal* if it has no induced cycle of length greater than k . Thus, chordal graphs are precisely the 3-chordal graphs. The problem of determining whether a graph is *k-chordal* is known to be co-NP-complete when k is a part of the instance [155].

³This class is introduced in [102], as the class of graphs $G = (V, E)$ with no chordless cycle of five or more vertices in G or in its complement $\overline{G} = (V, \overline{E})$.

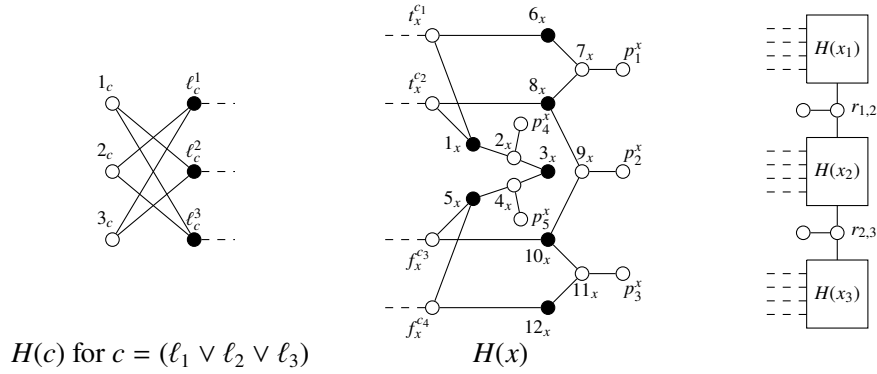


Figure 4.7: Clause gadget $H(c)$ and variable gadget $H(x)$ for EXT-CVC are shown on the left and in the middle of the figure respectively. Forced vertices (in U) are marked in black. On the right, the way of connecting variable gadgets is depicted. Crossing edges between $H(c)$ and $H(x)$ are marked with dashed lines.

Theorem 4.37. *EXT CVC is NP-complete in weakly triangulated graphs.*

Proof. We propose a simple reduction from $(3, B2)$ -SAT. Consider an instance $I = (\mathcal{C}, \mathcal{X})$ of $(3, B2)$ -SAT with clauses $\mathcal{C} = \{c_1, \dots, c_m\}$ and variables $\mathcal{X} = \{x_1, \dots, x_n\}$. We build a weakly triangulated graph $G = (V, E)$ together with a set of forced vertices U as follows:

- $V = Z \cup \{v_i, v'_i : i = 1, \dots, n\}$ where $Z = \{z_1, \dots, z_m\}$.
- The subgraph $G[Z]$ is a clique K_m and $v_i v'_i \in E$ for $i = 1 \leq n$.
- $z_j v_i \in E$ (resp. $z_j v'_i \in E$) if x_i appears positively (resp. negatively) in clause c_j and we set $U = Z$.

Property 4.38. *Graph G is weakly triangulated.*

Proof. By contradiction, we show that neither G nor \overline{G} possess an induced cycle of length at least 5. First suppose C is a chordless cycle containing at least 5 vertices in G . By considering the construction, C contains exactly 2 consecutive vertices of Z , because, on the one hand there is no cycle of length at least 4 in G containing at most one vertex from Z and on the other hand, if C contains at least three vertices of Z , by the structure of $G[Z]$, C has one chord. Hence, we can even conclude that $uv \in C$, since Z is a clique and then the length of C is at most four, contradiction.

Now, assume C is a chordless cycle of \overline{G} of length at least 5. By the construction of \overline{G} , in the subgraph induced by $V \setminus Z$, C either only contains consecutive vertices or at most 2 non consecutive vertices (corresponding to an edge G) because $\overline{G}(V \setminus Z)$ is isomorphic to $K_{2n} - nK_2$. If C contains at most 2 vertices from $V \setminus Z$, then C is of size at most 4 since Z is an independent set in \overline{G} , which is impossible. Otherwise, in $\overline{G}(V \setminus Z)$ the induced cycle C contains consecutive vertices and it is 3 vertices, x, y, z with $xz \in E$ (so, $\{xy, yz\} \subset C$). One more time, $|C| = 4$ which is a contradiction because Z is an independent set in \overline{G} . Therefore, G is weakly triangulated. \square

We here claim that $I = (\mathcal{C}, \mathcal{X})$ is satisfiable if and only if (G, U) is a *yes*-instance of EXT CVC . Actually, S is a minimal vertex cover including Z if and only if the truth assignment T satisfies all clauses of \mathcal{C} . In this regard, first suppose that T is a truth assignment of I , then we make a minimal connected vertex cover S of G containing Z such that $v_i \in S$ if $T(x_i) = \text{false}$ and $v'_i \in S$ if $T(x_i) = \text{true}$. Obviously, S is a connected vertex cover, because, all v_i and v'_i are connected to Z with two different edges and also S covers all edges of G . Moreover, since for each vertex in S there is at least one private edge to cover, then S is also minimal. If x_i appears negated (resp. unnegated) in clause c_j and satisfy this clause, then the edge $z_j v'_i$ (resp. $z_j v_i$) is a private edge to cover for z_j , and the edge $v_i v'_i$ is a private edge for v_i or v'_i to cover. Conversely, suppose S is a minimal connected vertex cover of G containing Z . By minimality, exactly one of v_i and v'_i for each $i \leq n$ is in S , so we make a truth assignment T such that $T(x_i) = \text{true}$ if $v'_i \in S$ and $T(x_i) = \text{false}$ if $v_i \in S$. By minimality of S , we can easily deduce that T satisfies all clauses of \mathcal{C} . \square

4.3.2 Price of extension

Regarding the concept *price of extension*, we define two optimization problems as follows:

$\text{EXT}_{\max}\text{CVC}$

Input: A connected graph $G = (V, E)$ and a set of vertices $U \subseteq V$.

Solutions: Minimal connected vertex cover S of G .

Output: Maximize $|S \cap U|$.

$\text{EXT}_{\min}\text{NSIS}$

Input: A connected graph $G = (V, E)$ and a set of vertices $U \subseteq V$.

Solutions: Maximal non separating independent set S of G .

Output: Minimize $|U \cup S|$.

For $\Pi = \text{EXT}_{\max}\text{CVC}$ or $\text{EXT}_{\min}\text{NSIS}$, we denote by $\text{opt}_{\Pi}(G, U)$ the value of an optimal solution. Since for both of them $\text{opt}_{\Pi}(G, U) = |U|$ if and only if (G, U) is a *yes*-instance of the extension variant, we deduce that $\text{EXT}_{\max}\text{CVC}$ and $\text{EXT}_{\min}\text{NSIS}$ are NP-hard since EXT CVC and EXT NSIS are NP-complete. Actually, for any class of graphs \mathcal{G} , $\text{EXT}_{\max}\text{CVC}$ is NP-hard in \mathcal{G} if and only if $\text{EXT}_{\min}\text{NSIS}$ is NP-hard in \mathcal{G} since for any graph $G \in \mathcal{G}$ it can be shown that:

$$\text{opt}_{\text{EXT}_{\max}\text{CVC}}(G, U) + \text{opt}_{\text{EXT}_{\min}\text{NSIS}}(G, V \setminus U) = |V|. \quad (4.2)$$

Adapted by Definition 2.4, we define the concept of *price of extension* for $\text{EXT}_{\max}\text{CVC}$ and $\text{EXT}_{\min}\text{NSIS}$ as follows:

Definition 4.39. For an instance (G, U) of $\text{EXT}_{\max}\text{CVC}$ and $\text{EXT}_{\min}\text{NSIS}$, $\text{PoE}(S)$ of a minimal connected vertex cover and a non separating independent set S of G is defined by $\text{PoE}(S) := \frac{|S \cap U|}{\text{opt}_{\text{EXT}_{\max}\text{CVC}}(G, U)}$ and $\text{PoE}(S) := \frac{|S \cup U|}{\text{opt}_{\text{EXT}_{\min}\text{NSIS}}(G, U)}$ respectively.

Using Propositions 4.32 and 4.33, we can derive negative and positive approximation results for $\text{EXT}_{\max}\text{CVC}$.

First, let us observe $\text{EXT}_{\min}\text{NSIS}$ does not admit $O(n^{1-\varepsilon})$ -PoE even in the simplest case $U = \emptyset$ because there is a simple reduction from MIN ISDS (which defined in Section 4.2.1) to $\text{EXT}_{\min}\text{NSIS}$ when $U = \emptyset$ by adding to the original graph $G = (V, E)$ two new vertices ℓ_0, ℓ_1 and edges $\ell_0\ell_1$ together with ℓ_1v for $v \in V$ (so, ℓ_1 is an universal vertex); ℓ_1 never belongs to a non separating independent set (or equivalently ℓ_0 is a part of all maximal non separating independent set) because otherwise ℓ_0 will become isolated. For general graphs, the price of extension associated to $\text{EXT}_{\max}\text{CVC}$ is hard to approximate.

Theorem 4.40. *For any constant $\varepsilon > 0$ and any $\rho \in \Omega\left(\frac{1}{\Delta^{1-\varepsilon}}\right)$ and $\rho \in \Omega\left(\frac{1}{n^{1-\varepsilon}}\right)$, $\text{EXT}_{\max}\text{CVC}$ does not admit a polynomial ρ -PoE for general graphs of n vertices and maximum degree Δ , unless $P = \text{NP}$.*

Proof. The proof uses the reduction given in Proposition 4.33 since actually it is an approximation preserving reduction from $\text{EXT}_{\max}\text{VC}$ to $\text{EXT}_{\max}\text{CVC}$. We have S is a minimal vertex cover of G and only if $S' = S \cup \{v'_1, \dots, v'_n\}$ is a minimal connected vertex cover of G' . Then, since $U' = U$, we deduce $|S' \cap U'| = |S \cap U|$. Finally, $n' = 3n$ and $\Delta(G') = \Delta(G) + 1$, together with lower bounds given in Corollary 4.24 give the expected result. \square

Although Proposition 4.33 preserves bipartiteness, we cannot immediately conclude the same kind of result since in Theorem 4.25, we proved that $\text{EXT}_{\max}\text{VC}$ admits a polynomial $\frac{1}{2}$ -PoE for bipartite graphs. The next theorem proposes a stronger lower bound.

Theorem 4.41. *For any constant $\varepsilon > 0$ and any $\rho \in \Omega\left(\frac{1}{n^{1/2-\varepsilon}}\right)$, $\text{EXT}_{\max}\text{CVC}$ does not admit a polynomial ρ -PoE for bipartite graphs of n vertices, unless $P = \text{NP}$.*

Proof. The proof is based on a gap reduction from MAX IS . Given a connected graph $G = (V, E)$ where $V = \{v_1, \dots, v_n\}$ and $E = \{e_1, \dots, e_m\}$, we build an instance (H, U) of $\text{EXT}_{\max}\text{CVC}$ as follows: $H = (V(H), E(H))$ is a connected bipartite graph together with a subset $U \subseteq V(H)$. First, we consider the incidence graph $G' = (V', E')$ of G where $V' = \{v_1, \dots, v_n\} \cup \{e_1, \dots, e_m\}$ and $E' = \{v_i e_k, v_j e_k : e_k = v_i v_j \in E\}$. Then, for each $v_i \in V'$, we add two vertices v_i^1, v_i^2 and connect them to v_i ; after that, we add a new vertex v_{n+1} and link it to $2n$ vertices $\{v_i^1, v_i^2 : 1 \leq i \leq n\}$; then, we add $m + n + 1$ new vertices p_1, \dots, p_{m+n+1} such that p_j is linked to e_j for $j = 1, \dots, m$ and p_{m+i} is linked to v_i^1 for $i = 1, \dots, n$. Finally, p_{m+n+1} is connected to v_{n+1} and $U = \{v_i^2 : i = 1, \dots, n\}$. An illustration of this construction is indicated in Figure 4.8.

Clearly H is a connected bipartite graph and it can be built in polynomial time. We now claim that, there is an independent set of size k of G if and only if a subset $U' \subseteq U$ of k vertices can be extended into a minimal connected vertex cover of H .

First of all, let us give some simple properties satisfied for each minimal connected vertex cover S of H . By minimality, S includes all non-leaf vertices incident to pendant edges of H , i.e. $\{e_1, \dots, e_m\} \cup \{v_1^1, \dots, v_n^1\} \cup v_{n+1} \subset S$. Also, by connectivity, $S' = S \cap \{v_1, \dots, v_n\}$ is a vertex cover of G . Furthermore, by minimality $v_i \in S$ if and only if $v_i^2 \notin S$.

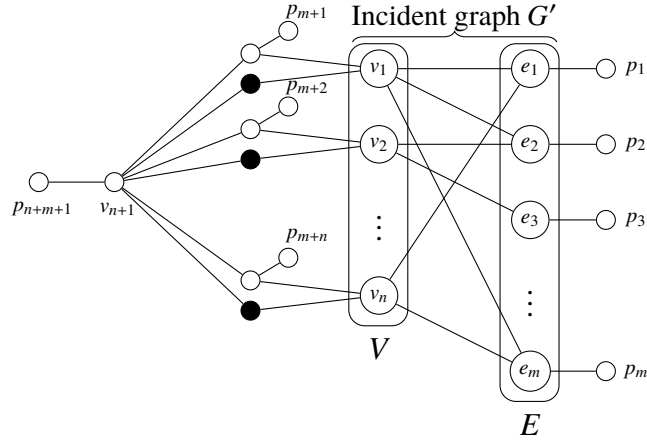


Figure 4.8: The graph $H = (V(H), E(H))$ built from $G = (V, E)$ with n vertices and m edges. Black vertices denote elements of U .

Hence, S is a minimal connected vertex cover of H with $|U \cap S| = k$ if and only if $V \setminus S'$ is an independent set of G with $|V \setminus S'| = k$. In conclusion:

$$\text{opt}_{\text{Ext}_{\max}\text{CVC}}(H, U) = \alpha(G) \quad (4.3)$$

Let $\varepsilon > 0$ be any constant. Given a graph $G = (V, E)$ on n vertices, by the hardness inapproximability result for MAX IS given in [165], there is no polynomial time algorithm distinguishing between the following two cases even when the graph is very dense, i.e. $m = n^{2-O(1/n)}$, unless $\text{NP} = \text{P}$:

- (Yes-Instance) $\alpha(G) \geq n^{\varepsilon-1}$,
- (No-Instance) $\alpha(G) < n^\varepsilon$.

Since in the worst case $|V(H)| = O(n^2)$, we get $\text{Ext}_{\max}\text{CVC}$ is not $O(\frac{1}{n^{1/2-\varepsilon}})$ -approximable in polynomial time for bipartite graphs on n vertices unless, $\text{P} = \text{NP}$. \square

We next present a positive result, showing that the price of extension is equal to 1 in chordal graphs.

Proposition 4.42. *$\text{Ext}_{\max}\text{CVC}$ is polynomial-time solvable in chordal graphs.*

Proof. Let (G, U) be an instance of $\text{Ext}_{\max}\text{CVC}$. We show that Proposition 4.32 can be adapted in the context of price of extension. Instead of adding a new pendent edge vv' for each cut-vertex v of G , we will add a new subtree $T_k(v)$ with special vertices $U'_v = \{v^1, \dots, v^k\}$ filled in black in the component $T_k(v)$ depicted on the left of Figure 4.9. After, the reduction proposed in Proposition 4.32 keeps the same, that is (G, U) instance of $\text{Ext}_{\max}\text{CVC}$ where $G = (V, E)$ is transformed into (G', U') where $U' = U \cup (V_c \setminus U) \cup U'_T$ where $U'_T = \cup_{v \in V_c} U'_v$ and $k = \Delta(G)$ (recall k is the number of leaves v^1, \dots, v^k of $T_k(v)$ and hence $|U'| = |U| + \Delta(G)|V_c| + |V_c \setminus U|$). We mainly prove the

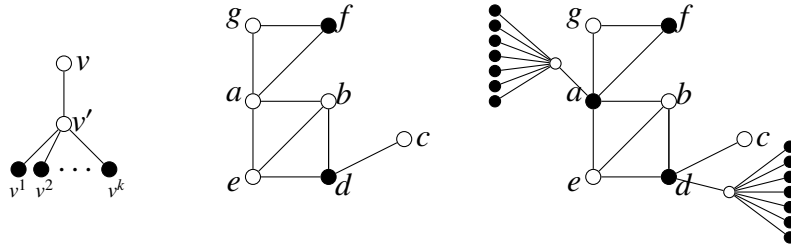


Figure 4.9: On the left side, tree $T_k(v)$ is depicted. Graph G and G' are drawn on the middle and on the right of the figure respectively. $V_c = \{a, d\}$ and $U = \{d, f\}$; forced vertices (i.e. U and U') are filled in black.

equality (4.4) given below which will conclude the proof since it is proved in Theorem 4.27 that $\text{EXT}_{\max} \text{VC}$ is linear-time solvable.

$$\text{opt}_{\text{EXT}_{\max} \text{VC}}(G', U') = \text{opt}_{\text{EXT}_{\max} \text{CVC}}(G, U) + \Delta(G)|V_c| + |V_c \setminus U| \quad (4.4)$$

Clearly, $\text{opt}_{\text{EXT}_{\max} \text{VC}}(G', U') \geq \text{opt}_{\text{EXT}_{\max} \text{CVC}}(G, U) + \Delta(G)|V_c| + |V_c \setminus U|$ because any minimal connected vertex cover S of G has to contain the whole cut-vertices set V_c of G . Hence, $S' = S \cup U'_T$ is a minimal vertex cover of G' with $|S' \cap U'| = |S \cap U| + \Delta(G)|V_c| + |V_c \setminus U|$.

Conversely, let S^{**} be an optimal minimal vertex cover of instance $I = (G', U')$ for $\text{EXT}_{\max} \text{VC}$. Inequality $\text{opt}_{\text{EXT}_{\max} \text{CVC}}(G, U) \geq \text{opt}_{\text{EXT}_{\max} \text{VC}}(G', U') - \Delta(G)|V_c| - |V_c \setminus U|$ comes from the assumption $U'_T \subseteq S^{**}$. Actually, we will prove the simple following claim:

Claim 4.43. $\forall v \in V_c$, either $\{v\} \cup U_v \subseteq S^{**}$ or $(\{v\} \cup U_v) \cap S^{**} = \emptyset$.

Proof. Let $v \in V_c$. We study two cases $v \in S^{**}$ and $v \notin S^{**}$. If $v \in S^{**}$, then we can assume $\{v^1, \dots, v^{\Delta(G)}\} \subseteq S^{**}$ since otherwise by minimality we would have $v' \in S^{**}$ and $S^{**} \cap \{v^1, \dots, v^{\Delta(G)}\} = \emptyset$. Flipping these two sets into S^{**} (i.e. adding $\{v^1, \dots, v^{\Delta(G)}\}$ and deleting $\{v'\}$) leads to another minimal vertex cover containing $\Delta(G)$ more vertices from U' . If $v \notin S^{**}$, then $S^{**} \cap U'_v = \emptyset$ by minimality. \square

From previous claim, assume by contradiction $S^{**} \cap U'_v = \emptyset$ (and then, $v' \in S^{**}$) for some $v \in V_c$. We prove that we can build another minimal vertex cover S' which is better than S^{**} . First, we flip these two sets (i.e. adding $\{v\} \cup U'_v$ and deleting $\{v'\}$); then, iteratively by deleting from S^{**} the neighbors of v which do not have a private edge, we get another minimal vertex cover S' of G' with $|S' \cap U'| \geq |S^{**} \cap U'| + |U'_v| + 1 - |N_U(v)| > |S^{**} \cap U'|$ which is impossible. Hence, the restriction S of S^{**} to G is a connected vertex cover with $|S \cap U| = \text{opt}_{\text{EXT}_{\max} \text{VC}}(G', U') + \Delta(G)|V_c| + |V_c \setminus U|$. In conclusion, equality (4.4) holds and the expected result follows. \square

Considering $\text{EXT}_{\max} \text{CVC}$ on $G = (V, E)$ in the particular case $U = V$, we obtain a new problem called UPPER CONNECTED VERTEX COVER (UPPER CVC for short) where the goal is to find the largest minimal connected vertex cover. To the best of our knowledge this problem has not been studied. In the next section we will study the (in)approximability of this problem.

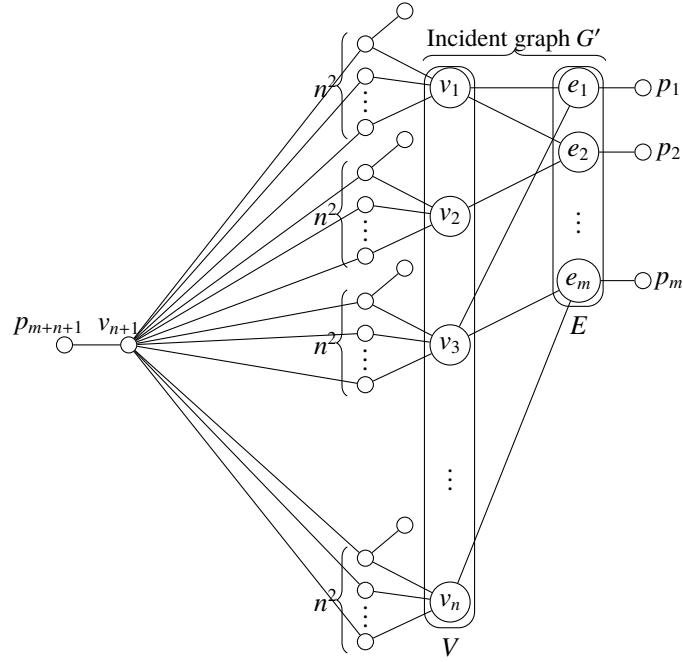


Figure 4.10: The graph $H = (V(H), E(H))$ built from $G = (V, E)$ with n vertices and m edges.

4.3.3 Upper CVC

Let start this section by a formal definition of UPPER CVC.

UPPER CVC

Input: A connected graph $G = (V, E)$.

Solutions: Minimal connected vertex cover $S \subseteq V$.

Output: Maximize $|S|$.

UPPER CVC is a special case of EXT_{\max} CVC where $U = V$. Regarding the approximability of UPPER CVC, we first show that an adaptation of Theorem 4.41 allows us to derive:

Corollary 4.44. *For any constant $\varepsilon > 0$, unless $\text{NP} = \text{P}$, UPPER CVC is not $\Omega(\frac{1}{n^{1/3-\varepsilon}})$ -approximable in polynomial time for bipartite graphs on n vertices.*

Proof. The reduction is similar to the one given Theorem 4.41 where we replace each vertex v_i^2 for $i = 2, \dots, n$ by the subset $\{v_i^2, \dots, v_i^{n^2}\}$. Each vertex v_i^ℓ for $i = 1, \dots, n$ and $\ell = 2, \dots, n^2$ is linked to v_i and v_{n+1} . Let H be the graph built. An illustration is given in Figure 4.10.

If $\text{opt}_{\text{UPPER CVC}}(H)$ denotes the optimal value of UPPER CVC on H , the equality (4.3) becomes:

$$\text{opt}_{\text{UPPER CVC}}(H) = 1 + 2n + m + \alpha(G)(n^2 - 2). \quad (4.5)$$

So, some simple calculus gives $n^2\alpha(G) \leq \text{opt}_{\text{UPPER CVC}}(H) \leq 2n^2\alpha(G)$ and $n^3 \leq |V(H)| \leq 2n^3$. In conclusion, we deduce it is hard to distinguish between two following cases:

- (Yes-Instance) $opt_{\text{UPPER CVC}}(H) \geq \frac{1}{2}|V(H)|^{1-\frac{\epsilon}{3}}$,
- (No-Instance) $opt_{\text{UPPER CVC}}(H) < 2|V(H)|^{\frac{2+\epsilon}{3}}$.

□

On the positive side, we show that any minimal connected vertex cover is a $\frac{2}{\Delta(G)}$ approximation for UPPER CVC. To do this, we first give a structural property that holds for any minimal connected vertex cover. For a given connected graph $G = (V, E)$ let S^* be an optimal solution of UPPER CVC and S be a minimal connected vertex cover of G . Denote by $A^* = S^* \setminus S$ and $A = S \setminus S^*$ the *proper* parts of S^* and S respectively, while $B = S \cap S^*$ is the *common* part. Finally, $R = V \setminus (S^* \cup S)$ denotes the *rest* of vertices. Also, for $X = A^*$ or $X = A$, we set $X_c = \{v \in X : N_G(v) \subseteq B\}$ which is exactly the vertices of X not having a neighbor in $(S \cup S^*) \setminus X$. Actually, $(S \cup S^*) \setminus X$ is either S or S^* .

Lemma 4.45. *The following properties hold:*

- (i) *For $X = A^*$ or $X = A$, $X \cup R$ is an independent set of G , $G[X \cup B]$ is connected and X_c is a subset of cut-set of $G[X \cup B]$.*
- (ii) *Set B is a dominating set of G .*

Proof. For (i). By construction, $X \cup R = V \setminus S$ or $V \setminus S^*$ and it is an independent set of G . Moreover, since $(\{A^*, A\}) \cup B = \{S^*, S\}$, then $G[X \cup B]$ is connected by hypothesis. Finally, if there is a vertex x of X_c which is not a cut-vertex of $G[X \cup B]$, then $(B \cup X) \setminus \{x\}$ remains a connected vertex cover of $G[X \cup B]$ which is a contradiction with minimality of $(B \cup X)$. Actually, we know vertices from X are only neighbors in B or in $(A^* \cup A) \setminus X$.

For (ii). We prove $V \setminus B \subseteq N_G(B)$. On the one hand, using Property (i), two sets $\{A^*, A\}$ are independent and since S and S^* induce connected subgraphs, then $(A^* \cup A) \subseteq N_G(B)$. On the other hand, since S^* and S are vertex covers then $R \subseteq N_G(B)$ (because otherwise one edge will be not covered by S^* or S). In conclusion, $V \setminus B = R \cup (A^* \cup A) \subseteq N_G(B)$ or equivalently B is a dominating set of G . □

The following theorem describes an interesting graph theoretic property. It relates the size of an arbitrary minimal connected vertex cover of a (connected) graph to the size of the largest minimal connected vertex cover.

Theorem 4.46. *Any minimal connected vertex cover of a connected graph G is a $\frac{2}{\Delta(G)}$ -approximation for UPPER CVC.*

Proof. Let $G = (V, E)$ be a connected graph. Let S and S^* be a minimal CVC and an optimal one for UPPER CVC, respectively, and w.l.o.g., assume $|S| < |S^*|$. We prove the following inequalities:

$$|A^*| \leq (\Delta(G) - 1)|B| \quad \text{and} \quad |A^*| \leq (\Delta(G) - 1)|A| \quad (4.6)$$

Let us prove the first part $|A^*| \leq (\Delta(G) - 1)|B|$ of inequality (4.6). Consider $v_1 \in B$ maximizing its number of neighbors in A^* , i.e. $v_1 = \arg \max\{|N_{A^*}(v)| : v \in B\}$. Since S is a minimal connected vertex cover with $|S| < |S^*|$, we have $\Delta(G) \geq |N_{A^*}(v_1)| + 1$ from (i) of Lemma 4.45 (otherwise $B = \{v_1\}$ with $d_G(v_1) = \Delta(G)$). In addition, from (ii) of Lemma 4.45 we have $N_{A^*}(B) = A^*$ and then $\sum_{v \in B} |N_{A^*}(v)| \geq |N_{A^*}(B)| = |A^*|$. Putting together these inequalities we get $|A^*| \leq |B|(\Delta(G) - 1)$.

Let us prove the second part $|A^*| \leq (\Delta(G) - 1)|A|$ of inequality (4.6) using the following claim:

Claim 4.47. There are at least $|A_c^*| + |A|$ edges between A and B in $G[S]$.

Proof. This claim come from connectivity of S and Lemma 4.45. First, we know each $v \in A$ has at least one neighbor in B . Moreover, since A_c^* is an independent set set only containing cut-vertices in $G[S^*]$, we deduce B has at least $|A_c^*| + 1$ connected components in $G[B]$. So there are at least $|A_c^*|$ other edges between A and B in $G[S]$ for linking the connected components in $G[B]$. \square

Each vertex in $A^* \setminus A_c^*$ has by definition at least one neighbor in A , so we deduce: $\sum_{v \in A} |N(v)| \geq |A^* \setminus A_c^*| + |A| + |A_c^*| = |A| + |A^*|$. Now, by setting $a_1 = \arg \max\{|N_G(v)| : v \in A\}$, we obviously get $|A||N(a_1)| \geq \sum_{v \in A} |N(v)|$. Putting together these inequalities, we obtain: $|A|\Delta(G) \geq |A||N(a_1)| \geq |A^*| + |A|$ which leads to $|A^*| \leq (\Delta(G) - 1)|A|$. The inequality $|S| \geq \frac{2}{\Delta(G)}$ follows by considering the two cases $|A| \geq |B|$ and $|A| < |B|$.

- Suppose $|A| \geq |B|$. By adding two inequalities $|A^*| \leq (\Delta(G) - 1)|A|$ and $|A^*| \leq (\Delta(G) - 1)|B|$ proved previously in inequalities (4.6) we get:

$$\begin{aligned} 2|A^*| &\leq (\Delta(G) - 1)|A| + (\Delta(G) - 1)|B| \\ 2|A^*| + 2|B| &\leq (\Delta(G) - 1)|A| + (\Delta(G) + 1)|B| \\ 2|S^*| &\leq \Delta(G)|S|. \end{aligned} \tag{4.7}$$

- Suppose $|A| < |B|$. The following ratio is increasing in $|B|$ since $|A^*| > |A|$. Then,

$$\begin{aligned} \frac{|S|}{|S^*|} &= \frac{|A| + |B|}{|A^*| + |B|} \geq \frac{|A| + |A|}{|A^*| + |A|} \\ &\geq \frac{2|A|}{(\Delta(G) - 1)|A| + |A|} = \frac{2}{\Delta(G)}. \end{aligned} \tag{4.8}$$

In any case, we conclude that $|S| \geq \frac{2}{\Delta}|S^*|$. Furthermore this ratio is tight for any $\Delta(G) \geq 3$. \square

A tight example of Theorem 4.46 for any $\Delta(G) \geq 3$ is illustrated in Figure 4.11. The optimal solution for UPPER CVC contains $\Delta(G)$ vertices $\{a\} \cup \{v_1, \dots, v_{\Delta(G)-1}\}$ while $\{a, b\}$ is a minimal connected vertex cover of size 2.

Corollary 4.48. UPPER CVC is APX-complete for bounded degree graphs.

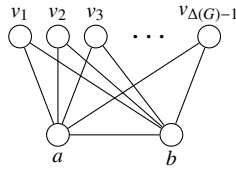


Figure 4.11: Graph G with $\Delta(G) + 1$ vertices and maximum degree $\Delta(G)$.

4.4 EXT DS

In this section, we consider extension variants of the classical graph problem DOMINATING SET. For a given graph $G = (V, E)$, a subset $S \subseteq V$ is called dominating set if for each vertex $v \in V$, v is in S or has at least one neighbor in S , i.e. $N[S] = V$. Here, for a given graph $G = (V, E)$ and a vertex set $U \subseteq V$, it is asked if there exists a *minimal* dominating set S with $U \subseteq S$. This problem formally defined as follows:

EXT DS

Input: A graph $G = (V, E)$ together with a forced set $U \subseteq V$.

Question: Is there a minimal dominating set D such that $U \subseteq D$?

We continue the section by proposing some hardness results for this problem.

4.4.1 Computational complexity

Here, we study hardness of EXT DS in general and some restricted graph classes.

Bipartite graphs

Theorem 4.49. *EXT DS is NP-complete on bipartite graphs of maximum degree 3, even if the subgraph $G[U]$ induced by the pre-solution U is an induced matching.*

Proof. The proof is based on a reduction from (3, B2)-SAT as defined before. For an instance I of (3, B2)-SAT with clause set $C = \{c_1, \dots, c_m\}$ and variable set $X = \{x_1, \dots, x_n\}$, we build a bipartite graph $G = (V, E)$ with maximum degree 3, together with a set $U \subseteq V$ of forced vertices as an instance of EXT DS as follows:

- For each clause $c = \ell_1 \vee \ell_2 \vee \ell_3$ where ℓ_1, ℓ_2, ℓ_3 are literals, we introduce the subgraph $H(c) = (V_c, E_c)$ with 7 vertices and 6 edges as illustrated on the left side of Figure 4.12. The vertices $1'_c$ and $2'_c$ represent literals in clause c ($1'_c$ represents literals ℓ_1 and ℓ_2 while $2'_c$ represents ℓ_3) and the vertices $\{3_c, 4_c\}$ are included in the forced set U_c .
- For each variable x , we introduce a gadget $H(x) = (V_x, E_x)$ which is a path of length 2. The vertex 1_x is in the set of forced vertices U_x . An illustration of variable subgraph $H(x)$ is on the right side of Figure 4.12.

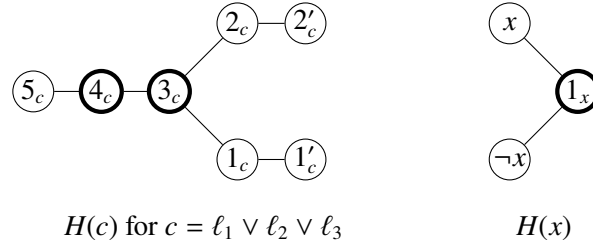


Figure 4.12: The Gadgets $H(c)$ and $H(x)$ for EXT DS. Vertices in U illustrated by their bold border.

- We interconnect the subgraphs $H(x)$ and $H(c)$ in the following way: for each clause c with literals ℓ_1, ℓ_2, ℓ_3 , corresponding to variables x_1, x_2, x_3 , respectively, connect $1'_c$ (representing ℓ_1 and ℓ_2) to the literal vertices ℓ_1 in $H(x_1)$ and ℓ_2 in $H(x_2)$ and connect $2'_c$ (representing ℓ_3) to the literal vertex ℓ_3 in $H(x_3)$.
- Also we set $U = (\bigcup_{c \in C} U_c) \cup (\bigcup_{x \in X} U_x)$.

This construction computes, in polynomial-time, a bipartite graph G with maximum degree 3. Moreover, $G[U]$ is an induced matching. In the following we can prove that there exists a satisfying assignment T for I if and only if (G, U) is a *yes*-instance of EXT DS.

Suppose T is a satisfying assignment for I . Create a dominating set S from U by adding for each $x \in X$ the literal vertex x if $T(x) = \text{true}$ and the literal vertex $\neg x$ if $T(x) = \text{false}$. Also, add to S for each clause $c \in C$ the vertex 1_c if $1'_c$ is not dominated by a literal vertex in S and 2_c if $2'_c$ is not dominated by a literal vertex in S . The resulting set S is obviously a dominating set for G which contains U . Since T satisfies all clauses in I , S does not contain both 1_c and 2_c for any clause c , so the vertex 3_c has at least one private neighbor for each $c \in C$. Since T further is a valid assignment, S only contains x or $\neg x$ for each variable $x \in X$, so 1_x also has a private neighbor. If S is not minimal, it can hence be turned into a minimal dominating set S' by successively removing vertices without private neighbors from the ones that we added to U . This could happen if there is a variable x whose setting does not matter, as all clauses that contain x or $\neg x$ are already satisfied by the other literals. Then, the corresponding literal vertex (x or $\neg x$) put into S can be removed. The resulting minimal dominating set S' still contains all vertices from U ; observe that vertices 4_c also have a private neighbor 5_c . The vertices 4_c and 5_c are present in the gadgets to prevent 3_c to consider itself as its private neighbor.

Conversely, assume that S is a minimal dominating set of G with $U \subseteq S$. Because of minimality, 3_c needs a private neighbor, either 1_c or 2_c . Hence, S contains at most one vertex from $\{1_c, 1'_c, 2_c, 2'_c\}$ for each clause c . In particular, there is at least one vertex among $\{1'_c, 2'_c\}$ which needs to be dominated by a literal vertex. Again by minimality, 1_x needs a private neighbor, either x or $\neg x$. Hence, the two literal vertices x and $\neg x$ cannot be together in S . Thus, by setting $T(x) = \text{true}$ (resp. false) if $x \in S$ (resp. $x \notin S$), we arrive at a partial truth assignment of I , satisfying all clauses, that can be easily completed. \square

Planar graphs

Here, we show that EXT DS is NP-complete in planar bipartite graphs of maximum degree 3. We present a new reduction, similar to what we did in Section 4.2 from 4P3C3SAT to EXT DS.

Theorem 4.50. *EXT DS is NP-hard for planar bipartite graphs of maximum degree 3.*

Proof. The proof is based on a reduction from 4P3C3SAT. For an instance I of 4P3C3SAT with clause set $\mathcal{C} = \{c_1, \dots, c_m\}$ and variable set $\mathcal{X} = \{x_1, \dots, x_n\}$, we build a planar bipartite graph $H = (V_H, E_H)$ with maximum degree 3 together with a set $U \subseteq V_H$ of forced vertices as an instance of EXT DS.

W.l.o.g. suppose that variable x_i appears in clauses c_1, c_2, c_3, c_4 of instance I such that in the induced (embedded) subgraph $G_i = G[\{x_i, c_1, c_2, c_3, c_4\}]$ of the graph BP defined in Definition 1.1, $c_1x_i, c_2x_i, c_3x_i, c_4x_i$ is an anti-clockwise ordering of edges around x_i . By looking at G_i and considering x_i appears positively and negatively, the construction should satisfy one of the following cases:

- case 1: $x_i \in c_1, c_2$ and $\neg x_i \in c_3, c_4$;
- case 2: $x_i \in c_1, c_3$ and $\neg x_i \in c_2, c_4$;
- case 3: $x_i \in c_1, c_2, c_3$ and $\neg x_i \in c_4$.

Note that all other cases are included in these 3 cases by rotations and/or replacing $x_i(\neg x_i)$ with $\neg x_i(x_i)$.

For each variable x_i , we propose 3 different gadgets $H(x_i)$ which are illustrated in Figure 4.13. As is depicted in the figure, the forced vertex set U_{x_i} corresponding to gadget $H(x_i)$ contains m_i for case 1, $\{p_i^1, p_i^2, m_i^1, m_i^2\}$ for case 2 and $\{p_i^1, p_i^2, p_i^3\}$ for case 3.

For each clause $c_j \in \mathcal{C}$, we consider a clause gadget $H(c_j)$ together with a forced vertex set U_{c_j} completely similar to what is defined before in Theorem 4.49 and illustrated in Figure 4.12. Moreover we interconnect with some crossing edges, the subgraphs $H(x_i)$ and $H(c_j)$ using the proposed instructions in Theorem 4.49. We also set the forced vertex set $U = (\bigcup_{x_i \in \mathcal{X}} U_{x_i}) \cup (\bigcup_{c_j \in \mathcal{C}} U_{c_j})$.

This construction computes in polynomial time, a planar bipartite graph with maximum degree 3. We now claim that (H, U) is a *yes*-instance of EXT DS if and only if I has a satisfying assignment T .

Suppose T is a truth assignment of I which satisfies all clauses. We construct a minimal dominating set S from U as follows:

- For each variable gadget $H(x_i)$ complies with "case 1", add t_i (resp. f_i) to S if $T(x_i) = \text{true}$ (resp., $T(x_i) = \text{false}$).
- For each variable gadget $H(x_i)$ complies with "case 2" add t_i^1, t_i^2 (resp. f_i^1, f_i^2) to S if $T(x_i) = \text{true}$ (resp., $T(x_i) = \text{false}$).

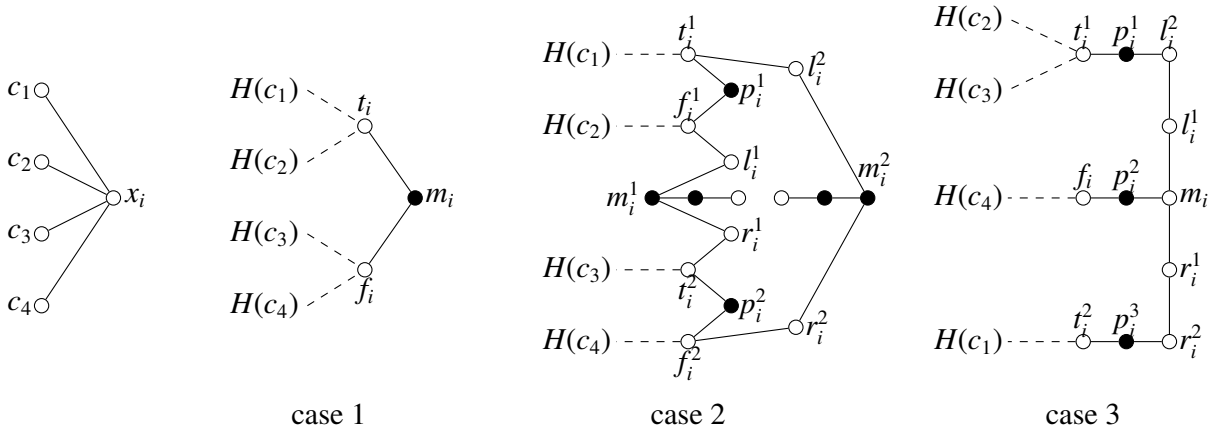


Figure 4.13: Variable gadgets $H(x_i)$ of Theorem 4.50. On the left: A variable x_i appearing in four clauses c_1, c_2, c_3, c_4 in I . On the right, cases 1, 2, 3 are corresponding to $H(x_i)$, depending on how x_i appears (negated or non-negated) in the four clauses (Here case 3 is rotated). Black vertices denote elements of U_{x_i} . Crossing edges are marked with dashed lines.

- For each variable gadget $H(x_i)$ complies with "case 3" add t_i^1, t_i^2, m_i (resp. f_i, l_i^2, r_i^2) to S if $T(x_i) = \text{true}$ (resp. $T(x_i) = \text{false}$).
- For each clause $c \in C$, add vertex 1_c to S if $1'_c$ is not dominated by a variable vertex of S and add 2_c to S if $2'_c$ is not dominated by a variable vertex of S .

Since T is satisfiable, for each clause gadget $H(c)$, at least one of $1'_c, 2'_c$ is dominated by a variable vertex of S . Thus, S is a dominating set that contains U of H . If S is not minimal, it can hence be turned into a minimal dominating set S' by successively removing vertices without private neighbors from the ones that we added to U .

Conversely, suppose S is a minimal dominating set of H with $U \subseteq S$. We show that in Theorem 4.49, for each clause gadget $H(c)$ at least one of $1'_c, 2'_c$ needs to be dominated by a variable gadget vertex thus there is an assignment T which satisfies all clauses of I . We now show that T is a valid assignment, in order to this, we consider all variable gadgets independently:

- If $H(x_i)$ complies with case 1, by minimality, S cannot contain both t_i, f_i . So we set $T(x_i) = \text{true}$ if $\{f_i\} \cap S = \emptyset$ and otherwise we set $T(x_i) = \text{false}$.
- If $H(x_i)$ complies with case 2, by minimality, S cannot contain both vertices in each pair $(t_i^1, f_i^1), (t_i^1, f_i^2), (t_i^2, f_i^1), (t_i^2, f_i^2)$, because we can remove the vertices $p_i^1, m_i^2, m_i^1, p_i^2$ respectively from S . So we set $T(x_i) = \text{true}$ if $S \cap \{f_i^1, f_i^2\} = \emptyset$ and otherwise we set $T(x_i) = \text{false}$.
- If $H(x_i)$ complies with case 3, by minimality, S cannot contain both vertices in each pair $(t_i^1, f_i), (t_i^2, f_i)$, because we can remove one of vertices in pairs $(p_i^1, p_i^2), (p_i^1, p_i^3)$ respectively from S . Hence, we set $T(x_i) = \text{true}$ if $S \cap \{f_i\} = \emptyset$ and otherwise, we set $T(x_i) = \text{false}$.

□

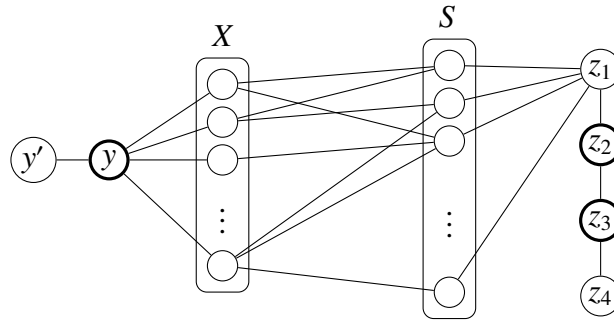


Figure 4.14: The graph $G = (V, E)$ for EXT DS, vertices in the forced set U' are drawn bold.

4.4.2 Parameterized complexity

We now study EXT DS in the framework of parameterized complexity where we consider the size of the set of forced vertices as *standard parameter* for our extension problems. We make a reduction from the extension variant of HITTING SET to EXT DS. HITTING SET as an NPO problem is defined by instances $I = (X, \mathcal{S})$ where X is a finite ground set and $\mathcal{S} = \{S_1, \dots, S_m\}$ is a collection of sets $S_i \subseteq X$ (usually referred to as *hyperedges*) and feasible solutions are subsets $H \subseteq X$ such that $H \cap S_i \neq \emptyset$ for all $i \in \{1, \dots, m\}$. In the optimization variant of the problem denoted by MIN HS, the goal is to find a subset H of minimum size. The extension variant of HITTING SET denoted by EXT HS defined formally as follows:

EXT HS

Input: A instance $I = (X, \mathcal{S})$ together with a forced set $U \subseteq X$.

Question: Is there an inclusion-wise minimal hitting set H such that $U \subseteq H$?

In [25], the extension variant of HITTING SET appears as a subproblem for the enumeration of minimal hitting sets in lexicographical order and EXT HS is there shown to be $W[3]$ -complete with respect to the standard parameter $|U|$. By a slight adjustment of the classical reduction from the MIN HS to MIN DS, this result transfers and formally yields:

Theorem 4.51. *EXT DS with standard parameter is $W[3]$ -complete, even when restricted to bipartite instances.*

Proof. EXT DS can obviously be modeled as special case of EXT HS by interpreting the closed neighborhoods as subsets of the ground set of vertices. This immediately gives membership in $W[3]$ for EXT DS.

Conversely, given an instance (I, U) with $I = (X, \mathcal{S})$, $\mathcal{S} = \{S_1, \dots, S_m\}$ for EXT HS we create a graph for the corresponding instance for EXT DS as follows:

- Start with the bipartite graph on vertices $X \cup \{s_1, \dots, s_m\}$ containing edges xs_i if and only if $x \in S_i$.

- Add two new vertices y, y' with edges $y'y$ and xy for all $x \in X$.
- Add four new vertices z_1, z_2, z_3, z_4 with edges z_1z_2, z_2z_3, z_3z_4 and z_1s_i for all $1 \leq i \leq m$.

The construction is illustrated in Figure 4.14. Let $G = (V, E)$ denotes the obviously still bipartite graph created in this way. With the set U' containing the vertex y to dominate X , z_2 and z_3 to forbid including any vertex s_i in the extension (as this would make z_2 obsolete) and the vertices corresponding to the forced set U for EXT HS, it is not hard to see that (G, U') is a *yes*-instance for EXT DS if and only if (I, U) is a *yes*-instance for EXT HS. As the parameters relate by $|U'| = |U| + 3$, this reduction transfers the W[3]-hardness of EXT HS to EXT DS on bipartite graphs. \square

4.4.3 Price of extension

Regarding the concept *price of extension* for EXT DS, we define a new optimization problem as follows:

EXT_{max}DS
Input: A graph $G = (V, E)$, a set of vertices $U \subseteq V$.
Solutions: Minimal dominating set S of G .
Output: Solution S that maximizes $|S \cap U|$.

We denote the value of optimal solution for an instance (G, U) of EXT_{max}DS, by $opt_{EXT_{max}DS}(G, U)$. Formally, it is $\arg \max\{U' \subseteq U : (G, U') \text{ is a } \textit{yes}\text{-instance of EXT VC}\}$.

Since $opt_{EXT_{max}DS}(G, U) = |U|$ if and only if (G, U) is a *yes*-instance of EXT DS, we deduce that EXT_{max}DS is NP-hard since EXT DS is NP-complete. Adapted by Definition 2.4, we define the concept of *price of extension* for EXT_{max}DS as follows:

Definition 4.52. For an instance (G, U) of EXT_{max}DS, $PoE(S)$ of a minimal dominating set S of G is defined by $PoE(S) := \frac{|S \cap U|}{opt_{EXT_{max}DS}(G, U)}$.

Dealing with EXT_{max}DS on graph $G = (V, E)$ and the particular forced set $U = V$, we obtain a well known optimization problem called *maximum minimal dominating set* or *Upper Dominating Set*, denoted by MAX-MIN DS. MAX-MIN DS has been studied extensively in recent years. In [137], it is shown that, the problem is NP-complete in Cubic bipartite graphs. Two inapproximability results $n^{\varepsilon-1}$ for graphs of order n and every $\varepsilon > 0$ and APX-hardness for cubic graphs are proposed in [18] for MAX-MIN DS. hence, we deduce:

Corollary 4.53. For any $\varepsilon > 0$, any $\rho \in \Omega(n^{\varepsilon-1})$, there is no polynomial-time ρ -approximation for EXT_{max}DS on general graphs of n vertices, unless $P = NP$.

In contrast to inapproximability result on the general graphs, we find that restriction to bipartite graphs or bounded degree graphs improves the approximability of EXT_{max}DS. If the instance is a bipartite graph. Since each part of the graph is a minimal dominating set, simply by selecting one side containing the largest number of vertices from the forced set U , we can show the following result:

Theorem 4.54. *A 2-approximation for $\text{Ext}_{\max}\text{DS}$ on bipartite graphs can be computed in polynomial time.*

Theorem 4.55. *A Δ -approximation for $\text{Ext}_{\max}\text{DS}$ on graphs of bounded degree Δ can be computed in polynomial time.*

Proof. Let (G, U) be an instance of $\text{Ext}_{\max}\text{DS}$, where $G = (V, E)$ is a graph of maximum degree Δ and $U \subseteq V$. If G is a complete graph on $\Delta + 1$ vertices, by minimality G cannot include more than one vertex of U , hence $\text{Ext}_{\max}\text{DS}$ is trivial. If $\Delta \leq 2$, a maximum independent set of $G[U]$, keeps at least half of the vertices of U , and can be extended to a minimal dominating set easily, so polynomially we have a $\frac{1}{2}$ -approximation algorithm for $\text{Ext}_{\max}\text{DS}$ in this case. Now, assume $\Delta \geq 3$ and $G \neq K_{\Delta+1}$. By Brooks's Theorem, we can color $G[U]$ properly with at most Δ colors in linear time. Let (S_1, \dots, S_ℓ) be such coloring with $\ell \leq \Delta$. Since each S_i is an independent set in $G[U]$, then (G, S_i) for $1 \leq i \leq \ell$ is a *yes*-instance of Ext DS . Thus, choosing $U' = \arg \max |S_i|$ yields a Δ -approximation. \square

4.5 Conclusion

In this chapter, we considered the extension variant of some vertex graph problems. Given a graph $G = (V, E)$ and a subset $U \subseteq V$ as an input of an anti-hereditary (resp. a hereditary) problem Π , it is asked, if there exists a minimal (resp. maximal) solution S such that $U \subseteq S$ (resp. $U \supseteq S$). Particularly, we considered the extension variant of Vertex Cover , Independent Set , Dominating Set , $\text{Connected Vertex Cover}$ and $\text{Non Separating Independent Set}$.

We have found many graph classes where the extension problems remains NP-complete, but also many classes where these problems are solvable in polynomial-time. Table 4.1 summarizes the complexity results obtained for the these problems.

| | NP-complete | Polynomial |
|---------------------|---|-------------------------------------|
| Ext VC/IS | sub-cubic planar bipartite graph cubic bipartite graph | chordal graph circular-arc graph |
| Ext CVC/NSIS | sub-cubic bipartite graph weakly triangulated graph | chordal graph |
| Ext DS | sub-cubic planar bipartite graph | |

Table 4.1: The computational complexity results

We also studied the parameterized complexity of these problems, with parameter $|U|$. All these complexity considerations are also carried out in very restricted scenarios, be it degree or

topological restrictions (bipartite, planar or chordal graphs). Table 4.2 summarizes the parameterised results obtained for these problems.

| | Ext VC | Ext IS | Ext CVC | Ext NSIS | Ext DS |
|---------------------------|---------------|---------------|----------------|-----------------|---------------|
| standard parameter | W[1]-com | FPT | W[1]-com | FPT | W[3]-com |
| dual parameter | FPT | W[1]-com | FPT | W[1]-com | FPT |

Table 4.2: The parameterised complexity results

We further discuss the *price of extension* (PoE), measuring the distance of U to the closest set that can be extended, which results in natural optimization problems related to extension problems for which we discuss polynomial-time approximability.

It would be also interesting to study further optimization problems that could be related to our extension problems, for instance, given an instance G, U, k of an anti-hereditary or a hereditary problem Π , is it possible to add at most k edges to the graph such that (G, U) becomes a *yes*-instance of $\text{Ext } \Pi$? Recall that adding edges among vertices from U does not change our problem, as they can never be private edges, but adding edges elsewhere might create private edges for certain vertices. Such problems would be defined according to the general idea of graph editing problems studied quite extensively in recent years. These problems are particularly interesting in graph classes where $\text{Ext } \Pi$ is solvable in polynomial time.

Extension of Some Edge Graph Problems

Contents

| | | |
|------------|---------------------------------------|------------|
| 5.1 | Introduction | 142 |
| 5.1.1 | Related work | 142 |
| 5.1.2 | Contributions | 143 |
| 5.2 | Ext EC and its generalization | 143 |
| 5.2.1 | Computational complexity | 144 |
| 5.2.2 | Parameterized complexity | 148 |
| 5.2.3 | Price of extension | 149 |
| 5.3 | Ext EDS and its generalization | 150 |
| 5.3.1 | Computational complexity | 151 |
| 5.3.2 | Parameterized complexity | 152 |
| 5.3.3 | Price of extension | 154 |
| 5.4 | Ext EM and its generalization | 156 |
| 5.4.1 | Computational complexity | 156 |
| 5.4.2 | Parameterized complexity | 162 |
| 5.4.3 | Price of extension | 164 |
| 5.5 | Conclusion | 165 |

Most of the results presented in this chapter will be presented in the following conference paper:

- Extension of some edge graph problems: standard and parameterized complexity. In Proceedings of the 22nd International Symposium on Fundamentals of Computation Theory, FCT 2019. Joint work with Katrin Casel, Henning Fernau, Jerome Monnot and Florian Sikora.

5.1 Introduction

In this chapter, we focus on edges extensions in graphs. More precisely, we consider *extension problems* related to several classical edge optimization problems in graphs, namely `EDGE COVER`, `MATCHING` and `EDGE DOMINATING SET`. Given a graph $G = (V, E)$ and an edge set $U \subseteq E$, it is asked whether there exists an (inclusion-wise) minimal (resp. maximal) feasible solution E' which satisfies a given property, for instance, being an edge dominating set (resp. an edge matching) and containing the *forced* edge set (resp. included in the *co-forbidden* edge set) U .

Let $G = (V, E)$ be a graph where the minimum degree is at least $r \geq 1$. For a fixed constant r :

- $S \subseteq E$ is called an *r-edge cover* of G , if each vertex $v \in V$ is incident to at least r edges of S . The special case $r = 1$ is known as an *edge cover*.
- $S \subseteq E$ is called an *r-edge dominating set* if any edge $e \in E$ is incident to at least r edges of S . The special case $r = 1$ is known as an *edge dominating set*.
- $S \subseteq E$ is called an *r-degree constrained partial subgraph* such that none of the vertices in V is incident to more than r edges in S . The special case $r = 1$ is known as an *edge matching*. (see Chapter 1 for the definition)

Actually, we consider the extension variants of the above edge graph problems in generalized version, when the constraint r is added to the problem's definition but not in the inputs of the instance. For example, for a given a graph $G = (V, E)$ with minimum degree at least r and an edge forced set $U \subseteq E$ as an instance of the extension version of *r-edge cover* denoted by `EXT r-EC`, it is asked whether there exists an (inclusion-wise) minimal *r-edge cover* E' containing the *forced* edge set U .

We present hardness results for these problems, we prove that these problems remain NP-complete, even in bipartite graphs of bounded degree and with some constraints on the forced/co-forbidden set of edges. Having a planar embedding does not help much either, as we show that these problems remain hard on subcubic bipartite planar graphs. We counter-balance these negative results with parameterized complexity results where we consider the size of the set of forced edges (resp. co-forbidden set) as standard parameter for the extension variant of `EDGE COVER` and `EDGE DOMINATING SET` (resp. `MATCHING`). We also consider the price of extension (`POE`), a natural optimization problem variant of extension problems, leading to some approximation results.

5.1.1 Related work

A well-studied generalization of `EDGE COVER`, known as the `MIN WEIGHTED LOWER-UPPER-COVER` which is already defined in Section 3.3. The `MIN WEIGHTED LOWER-UPPER-COVER` problem is known to be solvable in polynomial time (Chapter 35 in [151]). For the unweighted version i.e. $w(e) \in \{0, 1\}$, the goal is to minimize the cardinality $|M|$. Hence, an *r-edge cover* solution

corresponds to a lower-upper-cover with $a(v) = r$ and $b(v) = d_G(v)$ for every $v \in V$. The MINIMUM r -EDGE DOMINATING SET problem (MIN r -EDS for short) consists in finding an r -edge dominating set of minimum size for a given graph. The classical MIN EDGE DOMINATING SET problem (MIN EDS for short), is NP-hard in general graphs (problem [GT2] in [88]). The generalization to MIN r -EDS has been studied in [19, 20] (under the name b -EDS) from approximation point of view, when r is a part of the input. Two linear time approximation algorithms with ratios $\frac{8}{3}$ and 2 have been proposed for b -EDS problem in general graphs respectively in [19, 20]. In this latter case, the problem contains MIN EDS for particular case (corresponding to $r = 1$) and then is NP-hard. When r is a constant which is a part of the problem and not of the instance, the same approximation ratios can be derived, but not hardness result. However, to the best of our knowledge, MIN r -EDS for every $r \geq 2$ was not proved NP-hard so far. The optimization problem associated to r -degree constrained partial subgraph, denoted MAX r -DCPS, consists of finding an edge subset E' of maximum cardinality that is a solution to r -DCPS. It is also referred to as the Degree Constraint Subgraph Problem [85]. MAX r -DCPS is known to be solvable in polynomial time even for the edge weighted version (here, we want to maximize the weight of E') [85]. When additionally the constraint r is not uniform and depends on each vertex (i.e. at most $b(v) = r_v$, edges incident to vertex v), MAX r -DCPS is usually known as SIMPLE b -MATCHING and remains solvable in polynomial time even for the edge weighted version (Chapter 33 in [151]).

5.1.2 Contributions

We organize our results in this chapter as follows. In Section 5.2, the extension variant of EDGE COVER and its generalization is studied. We show that EXT r -EC for any $r > 1$ (resp. $r = 1$) is NP-hard in bipartite (resp. planar bipartite) graphs. In parameterized complexity, we show that EXT r -EC is in FPT respect to standard parameter, size of the forced set U . We further focus on the *Price of Extension* (PoE), and show that EXT_{max} r -EC in bipartite graphs is as hard as MAX IS in general graphs. We consider EXT r -EDS in Section 5.3, where we show that the problem is NP-hard for any $r \geq 1$ in planar bipartite graphs. We also show that the problem parameterized by the size of forced set U is W[1]-hard even in bipartite instances. Moreover, concerning PoE, we introduce a natural optimization problem corresponding to the EXT r -EDS and show that the problem in approximability is as hard as MAX IS in general graphs. We further in Section 5.4 focus on the extension variant of the well known problem EDGE MATCHING, The problem is NP-hard in bipartite graphs, and is in FPT with dual parameter $|\overline{U}|$. We also show that the optimization problem concerning PoE is in APX.

5.2 Ext EC and its generalization

In this section, we consider extension variant of the classical graph problems EDGE COVER and its generalization which is called r -EDGE COVER. A subset S of edges of the a graph is called edge cover if each vertex of graph is incident to at least one edge in S . Let $G = (V, E)$ be a graph where the minimum degree is at least $r \geq 1$. For a fixed constant r , S is called an r -edge cover of G , if each vertex $v \in V$ is incident to at least r edges of S . EXT EC has been partially

studied in Chapter 3 entitled "Extended Weighted Edge Cover Problem" in Section 3.2. Here, we complement these results to r -EC. To the best of our knowledge, the extension variant of r -EC has not been studied in the literature. We are considering the following *extension problem* associated to r -EDGE COVER.

EXT r -EC

Input: A graph $G = (V, E)$ and $U \subseteq E$.

Question: Does there exists $S \supseteq U$ such that the partial subgraph G_S is a minimal r -edge cover in G ?

We continue this section by proposing some hardness result.

5.2.1 Computational complexity

For our first NP-hardness proof we make reduction from $(3, B2)$ -SAT.

Theorem 5.1. *For every fixed $r \geq 1$, EXT r -EC is NP-complete in bipartite graphs with maximum degree $r + 2$, even if the forced edge set U is a matching.*

Proof. The proof is based on a reduction from $(3, B2)$ -SAT. A main building block $B_{2r}(v)$ (or $B_{2r+1}(v)$) in our construction is based on a complete bipartite subgraph of $2r$ (or $2r + 1$) vertices where one specified edge between two special vertices v and v' has been deleted. So, $B_{2r}(v) = K_{r,r} - \{vv'\}$ and $B_{2r+1}(v) = K_{r+1,r} - \{vv'\}$. Except for these two vertices v, v' , the other vertices of $B_{2r}(v)$ are not linked to any other vertex in the whole construction, while for $B_{2r+1}(v)$, it is only the case of v (i.e. only v is also linked outside $B_{2r+1}(v)$). Block $B_{2r}(v)$ will appear five times in each *variable gadget* $H(x)$, and block $B_{2r+1}(v)$ will correspond to each *clause gadget* (see Figure 5.1 for an illustration). By construction, all edges of $B_{2r}(v)$ will belong to any r -edge cover solution (in fact, vertices v and v' still need one more edge to satisfy the minimum degree constraint) and for $B_{2r+1}(v)$, it will be almost the case (except for few edges of $B_{2r+1}(v)$ incident to v , as all neighbors of v in $B_{2r+1}(v)$ have degree $r + 1$, and all edges between $N(v)$ and $N(N(v)) \setminus \{v\}$ have to be in the edge cover; v will need one more incident edge in the edge cover besides (some of) the edges from $B_{2r+1}(v)$).

Now, consider an instance I of $(3, B2)$ -SAT with clauses $\mathcal{C} = \{c_1, \dots, c_m\}$ and variables $\mathcal{X} = \{x_1, \dots, x_n\}$. We build a bipartite graph $G = (V, E)$ of maximum degree $r + 2$, together with a set U of permitted edges as follows:

- For each clause $c \in \mathcal{C}$, we build a clause gadget $B_{2r+1}(c)$ which is a component $K_{r,r-1}$ plus two vertices c, c' . An illustration of $B_{2r+1}(c)$ is given in the left side of Figure 5.1.
- For each variable $x \in \mathcal{X}$, we construct a subgraph $H(x) = (V_x, E_x)$ as follows: build two P_6 denoted $P = (x, l, m, r, \neg x)$ and $P' = (x', l', m', r', \neg x')$ respectively; then between each pair of vertices v, v' of P and P' a block $B_{2r}(v)$ is added for each v on P ; this interconnects v on P with the corresponding vertex v' on P' , as v and v' are special to $B_{2r}(v)$. The variable gadget $H(x) = (V_x, E_x)$ is illustrated to the right of Figure 5.1.

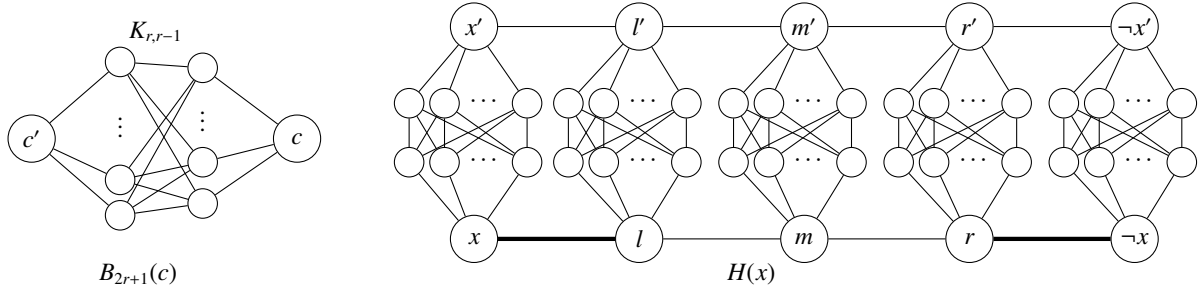


Figure 5.1: Block $B_{2r+1}(c)$ for clause c is depicted on the left-hand side. The subgraph $H(x) = (V_x, E_x)$ is shown on the right-hand side. Edges of U are drawn in bold.

- We interconnect $H(x)$ and $B_{2r+1}(c)$ where x is a literal of clause c by adding edge xc if x appears positively in c and the edge $\neg xc$ if x appears negated. Such edges will be called *crossing*.

Now, it is easy to see that G is bipartite of maximum degree $r+2$. Finally, let $U = \{x_i l_i, \neg x_i r_i : 1 \leq i \leq n\}$, picking the corresponding vertices and edges in each $H(x_i)$.

We claim that there is a truth assignment T of I which satisfies all clauses if and only if G admits a minimal r -edge cover solution $H = (V, S)$ where $U \subseteq S$ of G .

If T is a truth assignment of I which satisfies all clauses, a minimal r -EC solution $H = (V, S)$ can be constructed as follows:

- For each variable x , if $T(x) = \text{true}$, $\{xc : x \text{ appears positively in } c\} \cup (E_x \setminus \{lm, l'm'\}) \subseteq S$, and if $T(x) = \text{false}$, $\{\neg xc : x \text{ appears negatively in } c\} \cup (E_x \setminus \{mr, m'r'\}) \subseteq S$.
- Since T is a satisfying assignment, we have already added in the previous step $k \geq 1$ crossing edges connected to block $B_{2r+1}(c)$ for each clause c . Then, we delete arbitrarily $k-1$ edges S_c of $B_{2r+1}(c)$ incident to c , and we add $E(B_{2r+1}(c)) \setminus S_c$ to S .

Conversely, assume that $H = (V, S)$ is a minimal r -edge cover solution of G containing U . By considering the variable gadget $H(x)$ and in order to keep minimality S contains either lm or rm (not both since $\{xl, \neg xr\} \subset S$ by hypothesis and since all edges in the block $B_{2r}(v)$ for $v \in \{l, m, r\}$ have to be included into the edge cover by our previous observations). Hence, we set $T(x_i) = \text{true}$ if $rm \in S$ and $T(x_i) = \text{false}$ if $lm \in S$. Since H has to get a minimum degree at least r for each vertex and vertex c has a degree $r-1$ in clause gadget $B_{2r+1}(c)$, then the partial graph H contains at least one crossing edge incident to each c and hence T is a valid assignment of I . \square

In the following, we propose a reduction from 4P3C3SAT to EXT r -EC for the case that $r = 1$. The EXT 1-EC is same as EXT EC.

Theorem 5.2. *EXT 1-EC is NP-complete for planar bipartite subcubic graphs.*

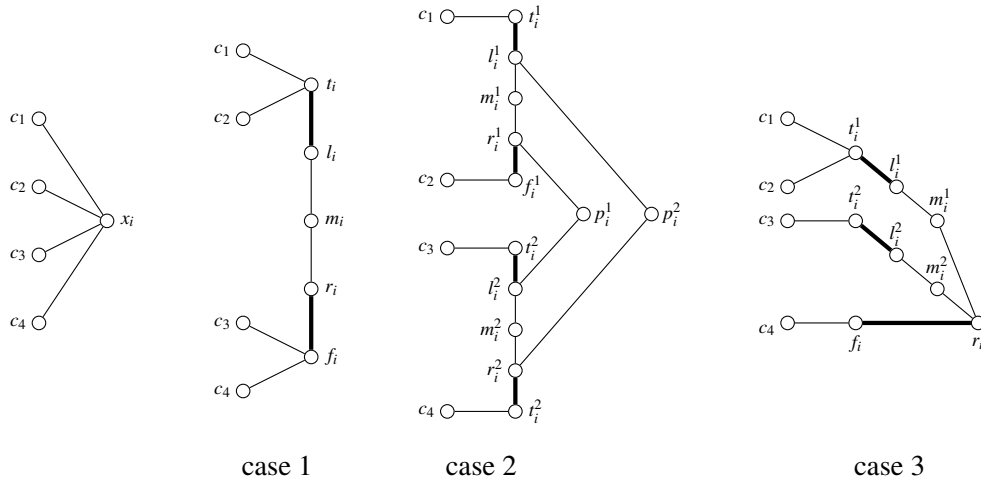


Figure 5.2: Construction for EXT 1-EC (planar). On the left: A variable x_i appearing in four clauses c_1, c_2, c_3, c_4 in I . On the right, cases 1, 2, 3: The gadgets $H(x_i)$ in the constructed instance depend on how x_i appears (negated or non-negated) in the four clauses. Bold edges denote elements of U .

Proof. The proof is based on a reduction from 4P3C3SAT. Let $I = (\mathcal{C}, \mathcal{X})$ be an instance of 4P3C3SAT, where $\mathcal{C} = \{c_1, \dots, c_m\}$ and $\mathcal{X} = \{x_1, \dots, x_n\}$ are the set of clauses and variables of I respectively. W.l.o.g. suppose that variable x_i appears in clauses c_1, c_2, c_3, c_4 of instance I such that in the induced (embedded) subgraph $G_i = G[\{x_i, c_1, c_2, c_3, c_4\}]$ of the graph BP defined in Definition 1.1, $c_1x_i, c_2x_i, c_3x_i, c_4x_i$ is an anti-clockwise ordering of edges around x_i . By looking at G_i and considering x_i appears positively and negatively, the construction should satisfy one of the following cases:

- case 1: $x_i \in c_1, c_2$ and $\neg x_i \in c_3, c_4$;
- case 2: $x_i \in c_1, c_3$ and $\neg x_i \in c_2, c_4$;
- case 3: $x_i \in c_1, c_2, c_3$ and $\neg x_i \in c_4$.

Note that all other cases are included in these by rotations or replacing x_i with $\neg x_i$ or vice versa. We illustrate how these cases are used in the reductions explicitly for EXT 1-EC. For each clause $c_i \in \mathcal{C}$, we consider a vertex c_i as a clause gadget. We connect each variable gadget $H(x)$ to the corresponding clause gadgets according differences to the cases listed above, see Figure 5.2.

We start from graph G and build a planar bipartite graph $H = (V_H, E_H)$ by replacing every node x_i in G with one of the three variable gadgets $H(x_i)$ which are illustrated in Figure 5.2. The forced edge set U_i , corresponding to variable gadget $H(x_i)$, contains $t_i l_i, r_i f_i$ for case (1), $t_i^1 l_i^1, r_i^1 f_i^1, t_i^2 l_i^2, r_i^2 f_i^2$ for case (2) and $t_i^1 l_i^1, t_i^2 l_i^2, r_i f_i$ for case (3). Let $U = \bigcup_{1 \leq i \leq n} U_i$, the set of forced edges of H . This construction can be done in polynomial time and the final graph H is planar bipartite with maximum degree 3. We now claim that I is satisfiable if and only if H admits a minimal edge cover containing U .

Suppose T is a truth assignment of I which satisfies all clauses. For each clause c_j , let $h(j)$ be an index such that variable $x_{h(j)}$ satisfies clause c_j for T and let $J = [n] \setminus h([m])$ be the unused indices by mapping h . We construct a minimal edge cover S of H by considering all possibilities of $H(x_i)$:

- for each variable gadget $H(x_i)$ which complies with "case (1)" we set:

$$\begin{aligned} S_1 := & \{t_{h(j)}c_j, m_{h(j)}r_{h(j)} : T(x_{h(j)}) = \mathbf{true}, x_{h(j)} \text{ appears positively in } c_j\} \\ & \cup \{f_{h(j)}c_j, m_{h(j)}l_{h(j)} : T(x_{h(j)}) = \mathbf{false}, x_{h(j)} \text{ appears negatively in } c_j\} \\ & \cup \{m_i r_i : i \in J\}. \end{aligned}$$

- for each variable gadget $H(x_i)$ which complies with "case (2)" by assuming $h(j) = k$ we set:

$$\begin{aligned} S_2 := & \{t_k^1 c_j (t_k^2 c_j), m_k^1 r_k^1, r_k^1 p_k^1, m_k^2 r_k^2, r_k^2 p_k^2 : T(x_k) = \mathbf{true} \wedge t_k^1 c_j \in E_H (t_k^2 c_j \in E_H)\} \\ & \cup \{f_k^1 c_j (f_k^2 c_j), m_k^1 l_k^1, l_k^1 p_k^1, m_k^2 l_k^2, l_k^2 p_k^2 : T(x_k) = \mathbf{false} \wedge f_k^1 c_j \in E_H (f_k^2 c_j \in E_H)\} \\ & \cup \{l_i^1 p_i^2, l_i^1 m_i^1, l_i^2 m_i^2, l_i^2 p_i^1 : i \in J\}. \end{aligned}$$

- for each variable gadget $H(x_i)$ which complies with "case (3)" by assuming $h(j) = k$ we set:

$$\begin{aligned} S_3 := & \{t_k^1 c_j (t_k^2 c_j), m_k^1 r_k, m_k^2 r_k : T(x_k) = \mathbf{true} \wedge t_k^1 c_j \in E_H (t_k^2 c_j \in E_H)\} \\ & \cup \{f_k c_j, m_k^1 l_k^1, m_k^2 l_k^2 : T(x_k) = \mathbf{false}\} \\ & \cup \{l_i^1 m_i^1, l_i^2 m_i^2 : i \in J\}. \end{aligned}$$

Finally we set $S = S_1 \cup S_2 \cup S_3 \cup U$. One can easily check that S is a minimal edge cover of H .

Conversely, suppose S is a minimal edge cover of H containing U . By minimality of S we propose an assignment T of I depending on different types of variable gadgets of H as follows:

- If $H(x_i)$ complies with case (1), in order to cover vertex m_i , the edge cover S either contains $m_i r_i$ or $m_i l_i$ (not both by minimality). This means that we set $T(x_i) = \mathbf{true}$ (resp. $T(x_i) = \mathbf{false}$) if $m_i r_i \in S$ (resp. $m_i l_i \in S$).
- If $H(x_i)$ complies with case (2), in order to cover vertices $m_i^1, m_i^2, p_i^1, p_i^2$, the edge cover S contains exactly one of edges in pairs $(l_i^1 m_i^1, r_i^1 m_i^1), (l_i^2 m_i^2, r_i^2 m_i^2), (l_i^1 p_i^2, r_i^2 p_i^2), (r_i^1 p_i^1, l_i^2 p_i^1)$. Hence, we set
 - $T(x_i) = \mathbf{true}$ if $\{l_i^1 m_i^1, l_i^1 p_i^2, l_i^2 p_i^1, l_i^2 m_i^2\} \cap S = \emptyset$, and
 - $T(x_i) = \mathbf{false}$ if $|\{l_i^1 m_i^1, l_i^1 p_i^2, l_i^2 p_i^1, l_i^2 m_i^2\} \cap S| \geq 1$.
- If $H(x_i)$ complies with 3, in order to cover vertices m_i^1, m_i^2 , S contains exactly one of edges in the pairs $(r_i m_i^1, l_i^1 m_i^1), (r_i m_i^2, l_i^2 m_i^2)$. This means that we set
 - $T(x_i) = \mathbf{false}$ if $S \cap \{m_i^1 r_i, m_i^2 r_i\} = \emptyset$, and
 - $T(x_i) = \mathbf{true}$ otherwise.

We obtain a valid assignment T . Since S covers all vertices of \mathcal{C} , T satisfies all clauses of I . \square

5.2.2 Parameterized complexity

We now study our problem in the framework of parameterized complexity. The next result is quite simple and characterizes the *yes*-instances of EXT r -EC.

Lemma 5.3. *(G, U) is a yes-instance of EXT r -EC if and only if there is an r -edge cover solution $G' = (V, E')$ where $E' \supseteq U$ such that $S_{G'} = \{v \in V(U) : d_{G'}(v) > r\}$ is an independent set of G_U .*

Proof. The condition is obviously necessary, as an edge among two vertices $x, y \in U$ of a minimal extension $X \supseteq U$ certifying that " (G, U) is a *yes*-instance", can only exist if x or y is, or both x and y are, incident to at most r edges from X because of minimality.

Let us look into the other direction. Let $G' = (V, E')$ be a partial subgraph of G with $U \subseteq E'$ and $d_{G'}(v) \geq r$ for all $v \in V$. Moreover, assume $S_{G'} = \{v \in V(U) : d_{G'}(v) > r\}$ is an independent set of G_U . Consider any minimal partial subgraph $H = (V, E_H)$ of $G' = (V, E')$ maintaining the property $d_{G'}(v) \geq r$ for all $v \in V$. Since $S_{G'}$ is an independent set of G_U , $U \subseteq E_H$ and therefore, E_H is a minimal r -edge cover containing U . \square

This structural property can be used to design an fpt-algorithm for EXT r -EC. More precisely, our proposed algorithm lists all $3^{|U|}$ many independent sets of $G[U]$ included in $V(U)$ from an instance $I = (G, U)$ of EXT r -EC. In each case, we produce an equivalent instance of MIN WEIGHTED LOWER-UPPER-COVER that can be solved in polynomial time which gives the following result.

Theorem 5.4. *EXT r -EC, with standard parameter, is in FPT.*

Proof. Let us establish a relation between the instances of the two problems EXT r -EC and MIN WEIGHTED LOWER-UPPER-COVER. Let (G, U) be a *yes*-instance of EXT r -EC where $G = (V, E)$ is a graph of minimum degree at least r and $U \subseteq E$. So, it implies by Lemma 5.3 the existence of a particular independent set S of G_U . We build an instance $(G_{\bar{U}}, a, b)$, $\bar{U} = E \setminus U$, of MIN WEIGHTED LOWER-UPPER-COVER, where a, b are two non-negative functions defined as follows:

$$a(v) := \begin{cases} r & \text{if } v \in V \setminus V(U) \\ r - d_{G_U}(v) & \text{if } v \in V(U), \end{cases}$$

and

$$b(v) := \begin{cases} d_G(v) & \text{if } v \in (V \setminus V(U)) \cup S \\ r - d_{G_U}(v) & \text{if } v \in V(U) \setminus S. \end{cases}$$

The next property is rather immediate.

Property 5.5. *If there is a solution of MIN WEIGHTED LOWER-UPPER-COVER for the instance $(G_{\bar{U}}, a, b)$, then (G, U) is a *yes*-instance of EXT r -EC.*

Proof. Assume that instance $(G_{\bar{U}}, a, b)$ of MIN WEIGHTED LOWER-UPPER-COVER admits a feasible solution and let $G^* = (V, E^*)$ be an optimal solution. Then, the partial graph $H = (V, E^* \cup U)$ satisfies the hypothesis of Lemma 5.3 (actually, H is already minimal with respect to property $\forall v \in V, d_H(v) \geq r$). \square

Using the outcome given in Theorem 5.5, the next result is rather straightforward. Consider the algorithm that lists all possible instances $(G[\overline{U}], a, b)$ for MIN WEIGHTED LOWER-UPPER-COVER by checking all independent sets of $G[U]$ included in $V(U)$ from an instance $I = (G, U)$ of EXT r -EC. This means that we try different values for function b . Since MIN WEIGHTED LOWER-UPPER-COVER is solvable in polynomial time [151], then the running time is dominated by the procedure that lists all possible independent sets of $G[U]$, i.e. there are $3^{|U|}$ possibilities: each vertex of each edge in U can be either included or excluded of the independent set, except for taking both endpoints in. \square

5.2.3 Price of extension

Regarding the concept *price of extension*, we define a new optimization problem in the following. Actually, since we mainly propose negative results, we only focus on the case $r = 1$:

EXT_{max}EC

Input: A connected graph $G = (V, E)$ and a set of edges $U \subseteq E$.

Solution: Minimal edge cover S of G .

Output: Maximize $|S \cap U|$.

For EXT_{max}EC, we denote the value of an optimal solution by $opt_{EXT_{max}EC}(G, U)$. Since $opt_{EXT_{max}EC}(G, U) \leq |U|$ with equality if and only if (G, U) is a *yes*-instance of the extension variant, we deduce from our previous results that EXT_{max}EC is NP-hard. Adapted by Definition 2.4, we define the concept of *price of extension* for EXT_{max}EC as follows:

Definition 5.6. For an instance (G, U) of EXT_{max}EC, PoE(S) of a minimal edge cover S of G is defined by $PoE(S) := \frac{|S \cap U|}{opt_{EXT_{max}EC}(G, U)}$.

We say that EXT_{max}EC admits a polynomial ρ -PoE if for every instance (G, U) , we can compute an edge cover S of G in polynomial time which satisfies $PoE(S) \geq \rho$.

Theorem 5.7. EXT_{max}EC in bipartite graph is as hard as MAX IS in general graphs.

Proof. The proof is based on a simple reduction from MAX IS. Given a graph $G = (V, E)$ with n vertices and m edges where $V = \{v_1, \dots, v_n\}$, as an instance of MAX IS, we build a connected bipartite graph $H = (V_H, E_H)$ as follows: for each $v_i \in V$, add a P_3 with edge set $\{v_i v'_i, v'_i v''_i\}$, and for each edge $e = v_i v_j \in E$ with $i < j$, add a middle vertex $v_{i,j}$ and connect v_i to v_j via $v_{i,j}$. Consider $I = (H, U)$ as instance of EXT_{max}EC, where the forced edge subset is given by $U = \{v_i v'_i : 1 \leq i \leq n\}$. Clearly, H is a bipartite graph with $|V_H| = 3n + m$ vertices and $|E_H| = 2(m + n)$ edges. The construction illustrated by a simple example in Figure 5.3. We claim that there is a solution of size k for EXT_{max}EC on (H, U) if and only if G has an independent set of size k .

Suppose that I is a maximal independent set of G of size k . For each $e \in E$, let $v^e \in V \setminus I$ be a vertex which covers e ; it is possible since $V \setminus I$ is a *vertex cover* of G . Moreover, $\{v^e : e \in E\} = V \setminus I$ since I is a maximal independent set of G . Clearly, $S = \{v_{i,j} v^e : e = v_i v_j \in E\} \cup \{v'_i v''_i : v_i \in V\} \cup \{v_i v'_i : v_i \in I\}$ is a minimal edge cover of H contains k edges of U .

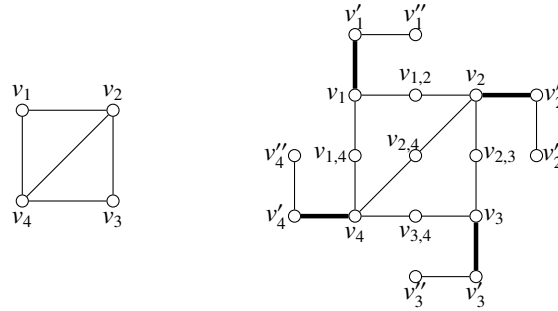


Figure 5.3: On the left side the graph G , an instance of MAX IS and on the left side, the graph H , the instance of EXT_{max}EC is shown. The bold edges in H are the set of forced edges U .

Conversely, suppose S is a minimal edge cover of H such that $|S \cap U| = k$. Let us make some simple observations of every minimal edge cover of H . Since, v'_i for $v_i \in V$ are leaves of H , so $\{v'_i v'_i : v_i \in V\}$ is a part of every edge cover. Moreover, for each $e = v_i v_j \in E$ with $i < j$, at least one edge between $v_i v_{i,j}$ or $v_j v_{i,j}$ belongs to any edge cover of H . Furthermore, if $v_i v_{i,j} \in S$, by minimality we deduce that $v_i v'_i \notin S$. Hence, for each $v_i v_j \in E$, at most one of $v_i v'_i, v_j v'_j$ can be in S . Hence, $I = \{v_i : v_i v'_i \in S\}$ is an independent set of G with size k . \square

Using the strong inapproximability results for MAX IS given in [154, 165], observing $\Delta(H) = \Delta(G) + 1$ and $|V_H| = 3|V| + |E| \leq 2|V|^2$, we deduce the following result.

Corollary 5.8. *For any constant $\varepsilon > 0$ and any $\rho \in \Omega(\Delta^{\varepsilon-1})$ and $\rho \in \Omega(n^{\varepsilon-\frac{1}{2}})$, EXT_{max}EC does not admit a polynomial ρ -PoE for general graphs of n vertices and maximum degree Δ , unless $P = NP$.*

5.3 Ext EDS and its generalization

An r -edge dominating set $S \subseteq E$ of a simple graph $G = (V, E)$ is a set S of edges such that for any edge $e \in E$ of G , at least r edges of S are incident to e (by definition, an edge dominates itself one time). The MINIMUM r -EDGE DOMINATING SET problem (MIN r -EDS for short) consists in finding an r -edge dominating set of minimum size. Obviously, MIN 1-EDS is the classical EDGE DOMINATING SET problem, which is NP-hard in general graphs (problem [GT2] in [88]). The generalization to MIN r -EDS has been studied in [19, 20] (under the name b -EDS) from approximation point of view. However, to the best of our knowledge, r -EDS for every $r \geq 2$ was not proved NP-hard so far. As associated extension problem, we formally study the following problem.

EXT r -EDS

Input: Given a simple graph $G = (V, E)$ and $U \subseteq E$.

Question: Is there a subset $S \subseteq E$ such that $U \subseteq S$ and S is a minimal r -edge dominating set?

We continue this section by proposing some hardness for Ext r -EDS for planar graphs.

5.3.1 Computational complexity

For the following NP-hardness result for EXT r -EDS we make reduction from 4P3C3SAT.

Theorem 5.9. *For any $r \geq 1$, EXT r -EDS is NP-complete for planar bipartite graphs of maximum degree $r + 2$.*

Proof. The proof is based on a reduction from 4P3C3SAT. Let $I = (\mathcal{C}, \mathcal{X})$ be an instance of 4P3C3SAT, where $\mathcal{C} = \{c_1, \dots, c_m\}$ and $\mathcal{X} = \{x_1, \dots, x_n\}$ are the set of clauses and variables of I respectively. W.l.o.g. suppose that variable x_i appears in clauses c_1, c_2, c_3, c_4 of instance I such that in the induced (embedded) subgraph $G_i = G[\{x_i, c_1, c_2, c_3, c_4\}]$ of the planar *vertex-clause-graph* G , $c_1x_i, c_2x_i, c_3x_i, c_4x_i$ is an anti-clockwise ordering of edges around x_i . By looking at G_i and considering x_i appears positively and negatively, the construction should satisfy one of the following cases:

- case 1: $x_i \in c_1, c_2$ and $\neg x_i \in c_3, c_4$;
- case 2: $x_i \in c_1, c_3$ and $\neg x_i \in c_2, c_4$;
- case 3: $x_i \in c_1, c_2, c_3$ and $\neg x_i \in c_4$.

Note that all other cases are included in these by rotations or replacing x_i with $\neg x_i$ or vice versa. In the following, we illustrate how these cases are used in the reductions explicitly for EXT r -EDS.

Consider first $r = 1$, corresponding to EXT EDS. We build a planar bipartite graph $H = (V_H, E_H)$ with maximum degree 3 together with a set $U \subseteq E_H$ of forced edges as an instance of EXT EDS. For each variable x_i we introduce a corresponding gadget $H(x_i)$ as depicted in Figure 5.4, the forced edge set U_{x_i} contains $\{m_i r_i, r_i p_i\}$ for case (1), $\{p_i^j r_i^j, r_i^j m_i^j : 1 \leq j \leq 4\}$ for case (2) and $\{p_i^1 p_i^2, p_i^2 p_i^3, p_i^5 p_i^6, p_i^6 p_i^7, m_i^2 f_i\}$ for case (3).

For each clause $c_j \in \mathcal{C}$, we construct a clause gadget $H(c_j)$ as depicted on the right side of Figure 5.5, and a forced edge set U_{c_j} , each clause gadget $H(c_j)$ contains 8 vertices and 7 edges where $|U_{c_j}| = 2$.

Moreover, we interconnect with some crossing edges the subgraphs $H(x_i)$ and $H(c_j)$ by linking x_i (or $\neg x_i$) to c_j according to how it appears in the clause. These crossing edges are incident to vertices $1'_c$ and $2'_c$ of clause gadget $H(c)$ in order to keep the planarity and by considering that the maximum degree remained 3. We also set the forced edge set $U = (\bigcup_{x_i \in \mathcal{X}} U_{x_i}) \cup (\bigcup_{c_j \in \mathcal{C}} U_{c_j})$. This construction is built in polynomial time, giving a planar bipartite graph of maximum degree 3. By considering the construction of clause gadget $H(c)$, by minimality at most one of the edges $1'_c 1_c, 2'_c 2_c$ can be a part of any minimal edge dominating set of H so, for each clause gadget $H(c)$ at least one of the crossing edges incident to $1'_c$ or $2'_c$ is contained in any minimal solution. On the other hand, by considering the 3 different variable gadgets $H(x)$, by minimality, one can see that for the case that the minimal solution S contains two crossing edges incident to a variable gadget $H(x)$, these edges have to be connected to the vertex (or vertices based on the $H(x)$) corresponding to positive appearance (t) or negative appearance (f), not both. By the above explanation, it is

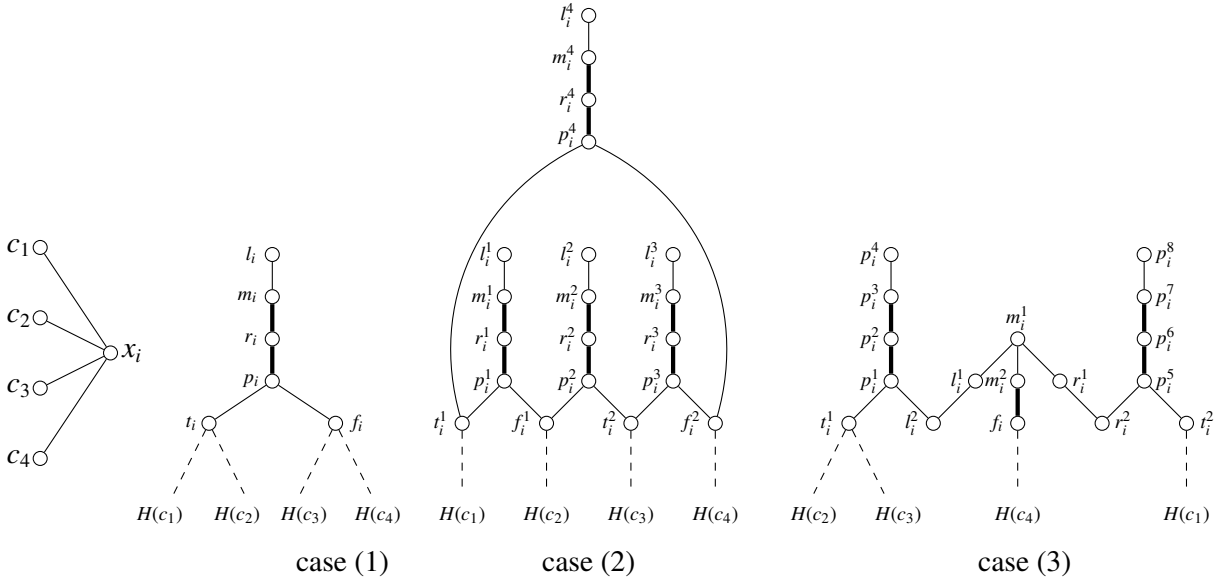


Figure 5.4: Variable gadgets $H(x_i)$ of Theorem 5.9. Cases (1), (2), (3) are corresponding to $H(x_i)$, depending on how x_i appears (as a negative or positive literal) in the four clauses (here, case 3 is rotated). Bold edges denote elements of U_{x_i} . Crossing edges are marked by dashed lines.

easy to see that I has a satisfying assignment T if and only if H has a minimal edge dominating set containing U .

For $r \geq 2$, we start with the instance $I = (H, U)$ given in the above construction for $r = 1$. Recall $H = (V_H, E_H)$ is a bipartite graph with bipartition $V_H = L \cup R$, while $U \subseteq E_H$ is a subset of forced edges. Now, for each vertex v of the left part L , we add the gadget $B_r(v)$ depicted to the left side of Figure 5.5. Denote by H' the resulting bipartite graph and consider $I' = (H', U)$ as an instance of EXT r -EDS. Let $B = \bigcup_{v \in L} B_r(v)$ be the added edges from H to H' . Note that any r -EDS S' of H' must contain B . Moreover, S' is a minimal r -EDS of H' if and only if $S' \setminus B$ is a minimal EDS of H . \square

5.3.2 Parameterized complexity

We now study EXT r -EDS in the framework of parameterized complexity.

Theorem 5.10. *For any $r \geq 1$, EXT r -EDS with standard parameter (size of forced set U) is $W[1]$ -hard, even when restricted to bipartite graphs.*

Proof. We only consider $r = 1$. For $r \geq 2$, we can use the gadget $B_r(v)$ as in Theorem 5.9. The hardness result comes from a reduction from EXT VC on bipartite graphs which proposed in Chapter 4. Let $I = (G, U)$ be an instance of EXT VC, where $G = (V, E)$ is a bipartite graph with partition (V_1, V_2) of V and $U \subseteq V$. We build an instance $I' = (G', U')$ of EXT EDS as follows.

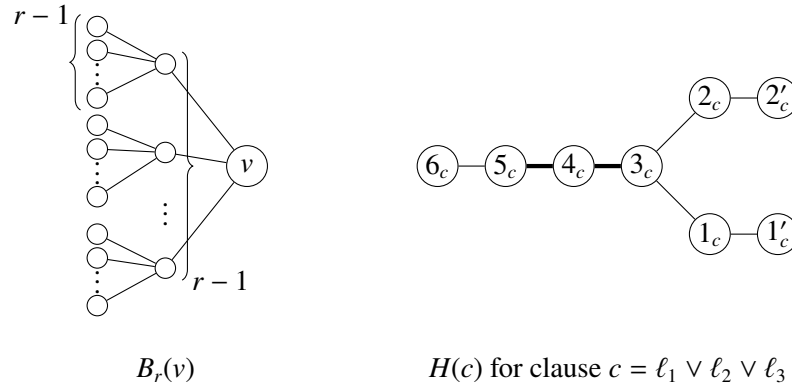


Figure 5.5: The component $B_r(v)$ and the clause gadget $H(c)$ of Theorem 5.9. In the clause gadget $H(c)$, forced edges are drawn in bold.

Let us first construct a new graph $G' = (V', E')$ with $V' = V \cup \{x_i, y_i, z_i : i = 1, 2\}$ and

$$E' = E \cup \bigcup_{i=1,2} (\{x_i y_i, y_i z_i\} \cup \{v x_i : v \in V_i\}).$$

G' is bipartite with partition into $V'_1 = V_1 \cup \{x_2, y_1, z_2\}$ and $V'_2 = V_2 \cup \{x_1, y_2, z_1\}$. Set $U' = \{u x_1 : u \in U \cap V_1\} \cup \{u x_2 : u \in U \cap V_2\} \cup \{x_1 y_1, x_2 y_2\}$ so, $|U'| = |U| + 2$. This construction is illustrated in Figure 5.6. We claim that (G', U') is a *yes*-instance of EXT EDS if and only if (G, U) is a *yes*-instance of EXT VC.

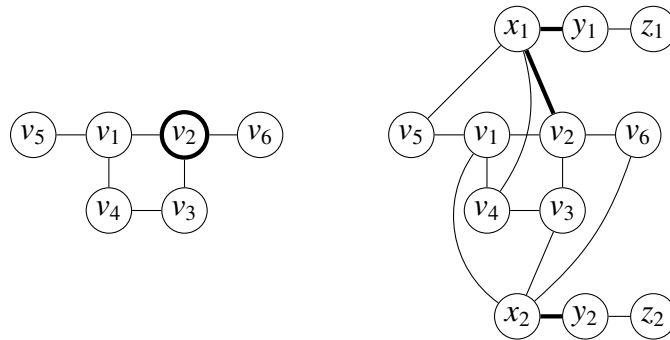


Figure 5.6: (G, U) as an instance of EXT VC is shown on the left, with $V_1 = \{v_2, v_4, v_5\}$ and $V_2 = \{v_1, v_3, v_6\}$ and $U = \{v_2\}$. The constructed instance (G', U') of EXT EDS is shown on the right. The vertices and edges of U and U' are in marked with bold lines.

Suppose (G, U) is a *yes*-instance for EXT VC; so there exists a minimal vertex cover S for G with $U \subseteq S$. The set $S' = \{v x_1 : v \in V_1 \cap S\} \cup \{v x_2 : v \in V_2 \cap S\} \cup \{x_1 y_1, x_2 y_2\}$ is an edge dominating set of G' which includes U' because S contains U . Since S is minimal, S' is minimal, too; observe that private edges of a vertex $v \in S \cap V_1$ translate to private edges of $v x_1 \in S'$, analogously for $x \in S \cap V_2$. By construction, $y_i z_i$ is a private edge for $x_i y_i$, $i = 1, 2$.

Conversely, suppose S' is a minimal edge dominating set of G' containing U' . Since S' is minimal, then for each $e \in S'$ there is a private edge set $S_e \subseteq E'$, $S_e \neq \emptyset$, which is dominated only by e . Moreover, we have, for $i \in \{1, 2\}$:

$$\forall v \in V_i ((vx_i \in S') \iff (\forall u \in V_{3-i} (vu \notin S' \cap E)))$$

since S' is minimal and $\{x_1y_1, x_2y_2\} \subseteq U'$. We now show how to safely modify S' such that $S' \cap E = \emptyset$. If it is not already the case, there is some edge, without loss of generality, $e = uv \in S' \cap E$ with $u \in V_1$ and $v \in V_2$. In particular from the above observations, we deduce $u \notin U$, $v \notin U$ and $S_e \subseteq E$. Modify S' by the following procedure.

- If the private solution set $S_e \setminus \{e\}$ contains some edges incident to u and some edges incident to v , then $e \in S'$ will be replaced by ux_1 and vx_2 ;
- if every edge in the private solution S_e is adjacent to u , replace e in S' by ux_1 , otherwise if every edge in the private solution S_e is adjacent to v , replace e in S' by vx_2 .

The case distinction is necessary to guarantee that S' stays a minimal edge dominating set after each modification step. We repeat this procedure until $S' \cap E = \emptyset$. At the end of the process, every vertex $v \in V$ covers the same set of edges as vx_1 or vx_2 dominates. Hence, by setting $S = \{v \in V: vx_1 \in S' \text{ or } vx_2 \in S'\}$, we build a minimal vertex cover of G containing U . \square

Remark 5.11. Note that the procedure of local modifications given in Theorem 5.10 does not preserve optimality, but only inclusion-wise minimality.

5.3.3 Price of extension

Regarding the concept *price of extension*, we define a new optimization problem in the following. Indeed, we only attend to the case $r = 1$ since we mainly propose negative results:

EXT_{max}EDS

Input: A connected graph $G = (V, E)$ and a set of edges $U \subseteq E$.

Solution: Minimal edge dominating set S of G .

Output: Maximize $|S \cap U|$.

For EXT_{max}EDS, we denote the value of an optimal solution by $opt_{\text{EXT}_{\max}\text{EDS}}(G, U)$. Since $opt_{\text{EXT}_{\max}\text{EDS}}(G, U) = |U|$ if and only if (G, U) is a *yes*-instance of the extension variant, hence, from our hardness result for EXT r -EDS, we deduce that EXT_{max}EC is NP-hard. Adapted by Definition 2.4, we define the concept of *price of extension* for EXT_{max}EDS as follows:

Definition 5.12. For an instance (G, U) of EXT_{max}EDS, PoE(S) of a minimal edge dominating set S of G is defined by $\text{PoE}(S) := \frac{|S \cap U|}{opt_{\text{EXT}_{\max}\text{EDS}}(G, U)}$.

We say that EXT_{max}EDS admits a polynomial ρ -PoE if for every instance (G, U) , we can compute an edge cover S of G in polynomial time which satisfies $\text{PoE}(S) \geq \rho$.

The next result is obtained by a simple approximation preserving reduction from EXT_{max}VC to EXT_{max}EDS.

Theorem 5.13. For any constant $\varepsilon \in (0, 1)$ and any $\rho \in \Omega(n^{\varepsilon-1})$, $\text{Ext}_{\max}\text{EDS}$ does not admit a polynomial ρ -PoE for general graphs of n vertices, unless $P = NP$.

Proof. Here, we propose a simple approximation preserving reduction from $\text{Ext}_{\max}\text{VC}$ to $\text{Ext}_{\max}\text{EDS}$ similar to Theorem 5.10. In Chapter 4, it is shown that $\text{Ext}_{\max}\text{VC}$ is hard to approximate within ratio $n^{\varepsilon-1}$ for any constant $\varepsilon \in (0, 1)$ for a graph of order n . Let $(G = (V, E), U)$ be an instance of $\text{Ext}_{\max}\text{VC}$, we build an instance $(G' = (V', E'), U')$ of $\text{Ext}_{\max}\text{EDS}$ such that $V' = V \cup \{v'\}$ and $E' = E \cup \{uv' : u \in V\}$. Let $U' = \{uv' : u \in U\}$. So G' is a graph with $|V'| = |V| + 1$ and $|U'| = |U|$.

We claim that there is a solution of size k for $\text{Ext}_{\max}\text{VC}$ on (G, U) if and only if $\text{Ext}_{\max}\text{EDS}$ has a solution of size k on (G', U') . Without loss of generality, suppose $U \neq \emptyset$.

Suppose S is a minimal vertex cover of G such that $|S \cap U| = k$, so $S' = \{uv' : u \in S\}$ is a minimal edge dominating set where $|S' \cap U'| = k$. Since S is a minimal vertex cover, so for any $u \in S$, there is at least a private edge $e_u \in E$ which is only covered by u , hence e_u will be a private edge for any $uv' \in S'$. Conversely, suppose S' is a minimal edge dominating set of G' containing k edges of U' . In the following, we will show that how we replace any minimal edge dominating set S' of G' with a new minimal solution S'' including just some edges incident to v' such that $S' \cap U' = S'' \cap U'$. The procedure is completely similar to what we did in Theorem 5.10, keeping all edges incident to v' in S' and replacing any edges non-incident to v' with some edges incident to v' as follows. For an non-incident edge $e = xy \in S'$, not incident to v' :

- if the private solution set $S_e \setminus \{e\}$ contains some edges incident to x and some edges incident to y , then $e \in S'$ will be replaced by xv' and yv' .
- if every edge in the private solution S_e is adjacent to x , replace e in S' by xv' , otherwise if every edge in the private solution S_e is adjacent to y , replace e in S' by yv' .

Now, by setting $S = \{u : uv' \in S''\}$, we make a minimal vertex cover of G , such that $|S'' \cap U'| = |S \cap U|$. Since $|V'| = |V| + 1$ and by considering the in-approximability result of $\text{Ext}_{\max}\text{VC}$ proposed in Chapter 4, the proof is completed. \square

In contrast to the last hardness result, we give a simple approximation depending on the maximum degree $\Delta(G)$.

Theorem 5.14. $\text{Ext}_{\max}\text{EDS}$ is $\frac{1}{\Delta(G_U)+1}$ -approximable for instance (G, U) of maximum degree Δ .

Proof. Let $(G = (V, E), U)$ be an instance of $\text{MExt}_{\max}\text{EDS}$, where the maximum degree of partial subgraph G_U induced by U is bounded by Δ . Compute a maximum matching M of G_U and transform it into a maximal matching M' of G containing M . It is well known that any maximal matching is an edge dominating set. Obviously, $(\Delta(G_U) + 1)|M| \geq |U| \geq \text{opt}_{\text{Ext}_{\max}\text{EDS}}(G, U)$ since G_U is $(\Delta(G_U) + 1)$ -edge colorable. \square

5.4 Ext EM and its generalization

Let $G = (V, E)$ be a graph where the minimum degree is at least $r \geq 1$. An r -degree constrained partial subgraph is defined as an edge subset $S \subseteq E$ such that none of the vertices in V is incident to more than r edges in S . The problem of finding such a set S of size at least k is termed r -DCPS. For the particular cases of $r = 1$, 1-DCPS corresponds to the famous EDGE MATCHING problem.

The optimization problem associated to r -DCPS, denoted MAX r -DCPS, consists of finding an edge subset E' of maximum cardinality that is a solution to r -DCPS. MAX r -DCPS is known to be solvable in polynomial time even for the edge weighted version [85, 151]. We are considering the following *extension problem* associated to r -DCPS.

EXT r -DCPS

Input: A graph $G = (V, E)$ and $U \subseteq E$.

Question: Does there exist $S \subseteq U$ such that the partial graph G_S has maximum degree at most r and is maximal in G ?

We continue this section by proposing some hardness results.

5.4.1 Computational complexity

For our first NP-hardness proof we make a reduction from (3, B2)-SAT.

Theorem 5.15. *For every fixed $r \geq 1$, EXT r -DCPS is NP-complete in bipartite graphs with maximum degree $\max\{3, r + 1\}$, even if \bar{U} is an induced matching for $r \geq 2$ or an induced collection of paths of length at most 2 for $r = 1$.*

Proof. We consider two different cases: $r = 1$ and $r \geq 2$.

First suppose $r = 1$. Consider an instance of (3, B2)-SAT with clauses $\mathcal{C} = \{c_1, \dots, c_m\}$ and variables $\mathcal{X} = \{x_1, \dots, x_n\}$. We build a bipartite graph $G = (V, E)$ of maximum degree 3 as follows:

- For each clause $c = x \vee y \vee z$, where x, y, z are literals, introduce a subgraph $H(c) = (V_c, E_c)$ with 8 vertices and 9 edges. V_c contains three specified vertices x_c, y_c and z_c corresponding to literals of the clause c . Moreover, $\bar{U}_c = \{x_c 1_c, y_c 2_c, z_c 3_c\}$ is a set of three forbidden edges included in $H(c)$. The gadget $H(c)$ is illustrated in the left part of Figure 5.7.
- For each variable x , introduce 12 new vertices. They induce the subgraph $H(x) = (V_x, E_x)$ illustrated in Figure 5.7. The vertex set V_x contains four special vertices $x^{c_1}, x^{c_2}, \neg x^{c_3}$ and $\neg x^{c_4}$, where it is implicitly assumed that variable x appears as a positive literal in clauses c_1, c_2 and as a negative literal in clauses c_3, c_4 . Finally, there are two sets of *free* edges (co-forbidden edges): $F_x = \{e_x\} \cup \{2_x^{c_3} \neg x^{c_3}, 2_x^{c_4} \neg x^{c_4}\}$ and $F_{\neg x} = \{e_{\neg x}\} \cup \{1_x^{c_1} x^{c_1}, 1_x^{c_2} x^{c_2}\}$. Hence, the forbidden edges U_x in $H(x)$ are given by $\bar{U}_x = E_x \setminus (F_x \cup F_{\neg x})$.

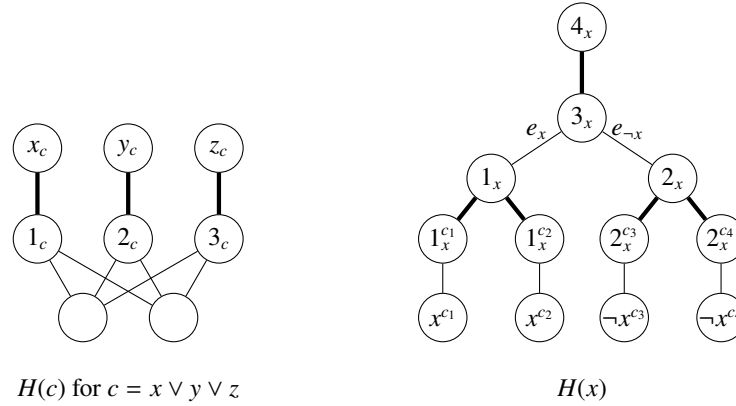


Figure 5.7: The Gadgets $H(c)$ and $H(x)$. Edges not in U are drawn as thicker lines.

- We interconnect $H(x)$ and $H(c)$, where x is a literal of clause c , by adding edge $x_c x^c$ if x appears as a positive literal and edge $x_c \neg x^c$ if x appears as a negative literal. We call these edges *crossing edges*.

We set $U = E \setminus (\bigcup_{c \in C} \overline{U}_c) \cup ((\bigcup_{x \in X} \overline{U}_x))$. This construction is computable within polynomial time and G is a bipartite graph of maximum degree 3. We claim that there is a truth assignment of I which satisfies all clauses if and only if there is a maximal matching $S \subseteq U$ of G .

If T is a truth assignment of I which satisfies all clauses, then we add the set of edges $x_c x^c$ and F_x if $T(x) = \text{true}$; otherwise, if $T(x) = \text{false}$, we add the edge $x_c \neg x^c$ and all edges in $F_{\neg x}$. For each clause c , we choose the literals which satisfy the clause (either one, two or three; let $\#c$ be the number of such literals); then, we add $3 - \#c$ edges saturating vertices $1_c, 2_c$ and 3_c . For instance, assume it is only y (thus, $\#c = 1$); then, we add two edges saturating vertices 1_c and 3_c and the white vertices in the gadget clause $H(c)$. The resulting matching S is maximal with $S \cap \overline{U} = \emptyset$.

Conversely, assume the existence of a maximal matching S with $S \subseteq U$. Hence, for each variable $x \in X$ exactly one edge between e_x and $e_{\neg x}$ belongs to S (in order to block edge $3_x 4_x$). If it is $e_x \in S$ (resp. $e_{\neg x} \in S$), then $F_x \subset S$ (resp. $F_{\neg x} \subset S$). Hence, S does not contain any crossing edges saturating $\neg x^c$ (resp. x^c). Now for each clause $c = x \vee y \vee z$, at least one vertex among x_c, y_c, z_c must be adjacent to a crossing edge of S . In conclusion, by setting $T(x) = \text{true}$ if at least one vertex x^{c1} or x^{c2} of $H(x)$ is saturated by S and $T(x) = \text{false}$ otherwise, we get a valid assignment T satisfying all clauses.

Assume now $r \geq 2$. The construction is an adaptation of the previous proof. A main building block of our construction is a subgraph, denoted $B_k(v)$ with $k < r$, containing $(kr) + 1$ vertices which are arranged as a tree of depth 2 with root v such that v has k children within this gadget and each child w of v has r children. For each child w of v one edge connecting w to a leaf in $B_k(v)$ will be forbidden in our construction, and we will use $F_{B_k(v)}$ to denote a fixed set of k edges in $B_k(v)$ such that each child of v is adjacent to an edge in $F_{B_k(v)}$ and v is not adjacent to an edge in $F_{B_k(v)}$. The left part of Figure 5.8 gives an illustration of $B_k(v)$. The purpose of this construction is that the root v will connect to other parts of the graph, and the structure of $B_k(v)$ with the forbidden

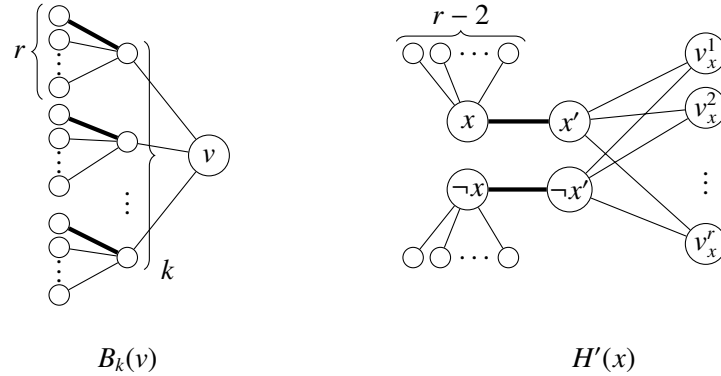


Figure 5.8: The gadgets $B_k(v)$ and $H'(x)$. Edges from the forbidden subset in $F_{B_k(v)}$ are marked with bold line of the left side and more generally, edges not in U are marked with bold line.

edges will make sure that a maximum r -degree constrained partial subgraph contains all edges between v and its children in $B_k(v)$, and can consequently only contain $r - k$ edges connecting v to a vertex outside $B_k(v)$. Namely, if one edge e would be missing from the edges incident to v in $B_k(v)$ in any maximal edge set E' , say, $e = vv'$, then E' would have to include the forbidden edge incident to v' by maximality.

Consider now an instance I of $(3, B2)$ -SAT with clauses $\mathcal{C} = \{c_1, \dots, c_m\}$ and variables $\mathcal{X} = \{x_1, \dots, x_n\}$. We build a bipartite graph $G = (V, E)$ of maximum degree $r + 1$, together with a set U of permitted edges (among which a maximal partial subgraph of degree at most r should be chosen) as follows:

- For each clause $c \in \mathcal{C}$, build a clause gadget $H(c) = (V_c, E_c)$ which is a $B_{(r-2)}(c)$ (the root c of $B_{(r-2)}(c)$ has $r - 2$ children). Hence, we denote $U_c = E_c \setminus F_{B_{(r-2)}(c)}$ set of permitted edges in $H(c)$.
- For each variable x introduce $3r$ new vertices which induce the primary subgraph denoted $H'(x) = (V'_x, E'_x)$. The vertex set V'_x contains four special vertices $x, x', \neg x, \neg x'$. The vertices x and $\neg x$ have $r - 2$ distinct vertices in its neighborhoods and x' and $\neg x'$ are connected to r common vertices $v_x^1, v_x^2, \dots, v_x^r$. Also we connect $x, \neg x$ to $x', \neg x'$ respectively with two forbidden edges in $H'(x)$. The right part of Figure 5.8 gives an illustration of $H'(x)$. By adding a component $B_{(r-1)}(y)$ for each vertex $y \in \{v_x^i : 1 \leq i \leq r\}$ and identifying the root of $B_{(r-1)}(y)$ with y , we construct a new subgraph $H(x) = (V_x, E_x)$. We define the set of forbidden edges in $H(x)$ by $F_x = \{xx', \neg x \neg x'\} \cup (\bigcup_{1 \leq i \leq r} F_{B_{(r-1)}(v_x^i)})$ and hence $U_x = E_x \setminus F_x$ denotes the set of permitted edges in $H(x)$.
- We interconnect $H(x)$ and $H(c)$ by adding edge xc if x appears positively in clause c and $\neg xc$ if x appears negatively. These *crossing edges* are always permitted and collected in the set U_{cross} .

Let $U = (\bigcup_{c \in \mathcal{C}} U_c) \cup (\bigcup_{x \in \mathcal{X}} U_x) \cup U_{\text{cross}}$ be the global set of forced edges. This construction is

computable in polynomial time, yielding a graph G that is a bipartite graph of maximum degree $r + 1$.

We claim that there is a truth assignment T of I which satisfies all clauses if and only if there is a maximal r -degree constrained partial subgraph $G_S = (V, S)$ where $S \subseteq U$ of G .

If T is a truth assignment of I which satisfies all clauses, a maximal r -degree constrained partial subgraph $G_S = (V, S)$ with $S \subseteq U$ can be constructed as follows:

1. For each variable gadget $H(c)$, by maximality $U_c \subseteq S$.
2. For each variable x we add edges according to the assignment as follows: If $T(x) = \text{true}$, we add $v_x^i x'$ for all $1 \leq i \leq r$ and the two crossing edges connecting $\neg x$ with their respective clause gadgets. If $T(x) = \text{false}$, we add $v_x^i \neg x'$ for all $1 \leq i \leq r$ and the two crossing edges connecting x with their respective clause gadgets. In both cases by maximality we add $2(r - 2)$ pendent edges incident to x and $\neg x$ and also all permitted edges in $B_{(r-1)}(y)$ for all $y \in \{v_x^i : 1 \leq i \leq r\}$.
3. At last, for each variable c which has more than one true literal in assignment T , add some arbitrary crossing edges to c such that $d_{G_S}(c) = r$.

The resulting subgraph is a maximal r -degree constrained partial subgraph $G_S = (V, S)$ with $S \subseteq U$.

Conversely, assume the existence of a maximal r -degree partial subgraph $G_S = (V, S)$ with $S \subseteq U$. First, recall that for every gadget $B_k(v)$, we must have $E(B_k(v)) \setminus F_{B_k(v)} \subseteq S$ for $k \in \{r - 1, r - 2\}$. Moreover, for each variable gadget $H(x)$, at least one of the pairs of crossing edges incident to x and $\neg x$ have to be in S (by maximality). Hence we set $T(x) = \text{true}$ if both crossing edges incident to $\neg x$ are in S and otherwise we set $T(x) = \text{false}$ (if both, we choose arbitrarily one of them). This assignment is valid and since for each clause c , at most two crossing edges incident to vertex c are in S (G_S is a subgraph with maximum degree r), then T satisfies all clauses of I . \square

In Theorem 5.15, we showed that, for every fixed $r \geq 2$, EXT r -DCPS is hard even when the set of forbidden edges $E \setminus U$ is an induced matching. In the following, we prove the same result does not hold when $r = 1$, by reducing this problem to the problem of finding a maximum matching in a bipartite graph.

Proposition 5.16. *EXT 1-DCPS is polynomial-time decidable when the forbidden edges $\overline{U} = E \setminus U$ form an induced matching.*

Proof. Here, we reduce this problem to the problem of finding a maximum matching in a bipartite graph. Let $G = (V, E)$ along with $U \subseteq E$ be an instance of EXT 1-DCPS where the partial graph $G_{\overline{U}}$ induced by $E \setminus U$ is a collection of disjoint edges and isolated vertices. Consider the bipartite graph $G' = ((L \cup R), E')$ build as follows: the vertex set is $L = \{l_e : e \notin U\}$ and $R = \{r_v : v \in V \setminus V(E \setminus U)\}$; $l_e r_v \in E'$ if and only if $v x \in E$ or $v y \in E$ with $e = xy \notin U$.

Now, we claim that, (G, U) is a *yes*-instance of EXT 1-DCPS if and only if there is a matching M' of G' with $|M'| = |L|$.

The condition is trivially sufficient because a matching M' of G' with $|M'| = |L|$ corresponds to a matching M of G which is adjacent to every edge of $E \setminus U$. Hence, by considering any maximal matching S of G which extend M , we get $S \subseteq U$. The condition is necessary. Suppose S is a maximal matching of G such that $S \subseteq U$; this implies that every edge $e = xy \in E \setminus U$ is adjacent to some edge $e' = xv$ (or yv) of S . Since $G_{\overline{U}}$ is an induced matching, $v \notin V(E \setminus U)$ or equivalently $l_e r_v \in E'$. Hence, this subset of edges $l_e r_v \in E'$ for $e \in (E \setminus U)$ forms a matching M' in G' saturating all vertices in L . \square

Remark 5.17. Proposition 5.16 can be extended to the case where \overline{U} is a matching and $G_{\overline{U}}$ does not contain an alternating path of length at least 5. The complexity of EXT 1-DCPS when \overline{U} is a matching remains unsettled.

In Chapter 4, several results are proposed for EXT IS and EXT VC in bipartite graphs, planar graphs, chordal graphs, etc. Here, we deduce a new result for a subclass of claw-free graphs.

Corollary 5.18. *EXT IS is NP-complete restricted to line graphs of bipartite graphs of maximum degree 3.*

Proof. Let $G = (V, E)$ be a bipartite graph of maximum degree 3 and $L(G) = (V', E')$ its line graph. It is well known that any matching S of G corresponds to an independent set $S' = L(S)$ of G' and vice versa. In particular, S is a maximal matching of G if and only if $L(S)$ is a maximal independent set. Hence, (G, U) is a *yes*-instance of EXT 1-DCPS if and only if $(L(G), L(U))$ is a *yes*-instance of EXT IS. Theorem 5.15 with $r = 1$ concludes the proof. \square

Corollary 5.19. *EXT VC is NP-complete restricted to line graphs of bipartite graphs of maximum degree 3.*

In the following, we propose a reduction from 4P3C3SAT to EXT r -DCPS for the case that $r = 1$. The EXT 1-DCPS is same as EXT EM.

Theorem 5.20. *EXT 1-DCPS is NP-complete for planar bipartite subcubic graphs.*

Proof. The proof is based on a reduction from 4P3C3SAT. Let $I = (\mathcal{C}, \mathcal{X})$ be an instance of 4P3C3SAT, where $\mathcal{C} = \{c_1, \dots, c_m\}$ and $\mathcal{X} = \{x_1, \dots, x_n\}$ are the set of clauses and variables of I respectively. W.l.o.g. suppose that variable x_i appears in clauses c_1, c_2, c_3, c_4 of instance I such that in the induced (embedded) subgraph $G_i = G[\{x_i, c_1, c_2, c_3, c_4\}]$ of the graph BP defined in Definition 1.1, $c_1 x_i, c_2 x_i, c_3 x_i, c_4 x_i$ is an anti-clockwise ordering of edges around x_i . By looking at G_i and considering x_i appears positively and negatively, the construction should satisfy one of the following cases:

- case 1: $x_i \in c_1, c_2$ and $\neg x_i \in c_3, c_4$;
- case 2: $x_i \in c_1, c_3$ and $\neg x_i \in c_2, c_4$;

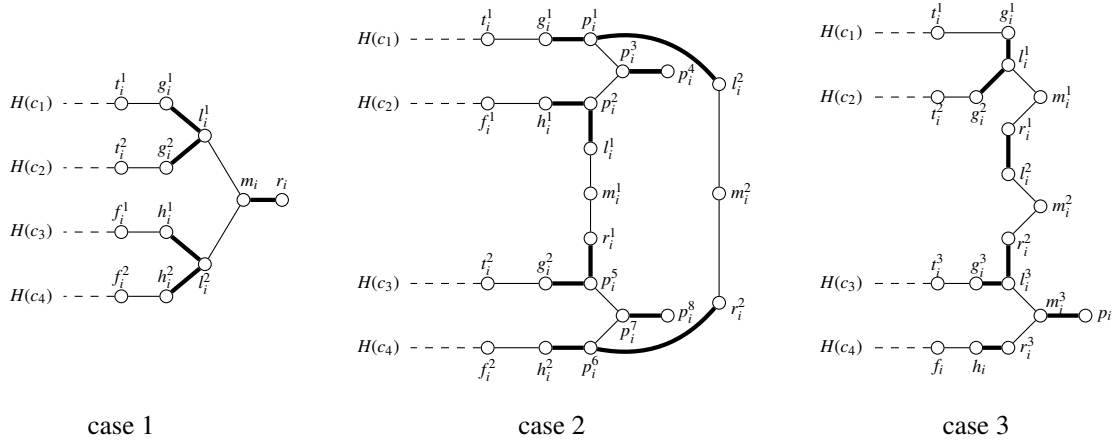


Figure 5.9: Variable gadgets $H(x_i)$ of Theorem 5.20. Cases (1), (2), (3) are corresponding to $H(x_i)$, depending on how x_i appears (negated or non-negated) in the four clauses. Edges not in U_{x_i} are drawn bold. Crossing edges are marked with dashed lines.

- case 3: $x_i \in c_1, c_2, c_3$ and $\neg x_i \in c_4$.

Note that all other cases are included in these by rotations or replacing x_i with $\neg x_i$ or vice versa.

We build a planar bipartite graph $H = (V_H, E_H)$ with maximum degree 3 together with a set $U \subseteq E_H$ of permitted edges as an instance of EXT 1-DCPS as follows.

- For each clause c_j , we introduce a clause gadget $H(c_j)$ together with a permitted edge set U_{c_j} which is already explained in detail in Theorem 5.15 for $r = 1$.
- For each variable x_i depending on how x_i appears (negated or non-negated) in clauses, we introduce 3 different gadgets $H(x_i)$ together with a set of permitted edges U_{x_i} which is depicted in Figure 5.9.
- We also interconnect $H(x_i)$ to $H(c_j)$, where x_i appears positively or negatively in clause c_j by crossing edges like in the proof of Theorem 5.15. Let U_{cross} be the set of all crossing edges.

Let $U = (\bigcup_{c_j \in \mathcal{C}} U_{c_j}) \cup (\bigcup_{x_i \in \mathcal{X}} U_{x_i}) \cup U_{cross}$. This construction computes in polynomial time a planar bipartite graph of maximum degree 3. We now claim that (H, U) is a *yes*-instance of EXT 1-DCPS if and only if T is a satisfying assignment of I .

Suppose T is a truth assignment of I which satisfies all clauses. We produce a maximal edge matching $S \subseteq U$ as follows: the method of choosing edges from clause gadgets and crossing edges is already explained in Theorem 5.15 for $r = 1$, so we here just show which edges of each $H(x_i)$ should be in S :

- If $H(x_i)$ complies with case (1), we add $\{m_i l_i^1, h_i^1 f_i^1, h_i^2 f_i^2\}$ (resp. $\{m_i l_i^2, g_i^1 t_i^1, g_i^2 t_i^2\}$) if $T(x_i) = \text{true}$ (resp. $T(x_i) = \text{false}$),

- If $H(x_i)$ complies with case (2), we add $\{f_i^1 h_i^1, p_i^1 p_i^3, m_i^2 r_i^2, f_i^2 h_i^2, p_i^5 p_i^7, l_i^1 m_i^1\}$ if $T(x_i) = \text{true}$; if $T(x_i) = \text{false}$, we add $\{p_i^2 p_i^3, t_i^1 g_i^1, l_i^2 m_i^2, p_i^6 p_i^7, t_i^2 g_i^2, m_i^1 r_i^1\}$,
- If $H(x_i)$ complies with case (3), we add $\{l_i^1 m_i^1, l_i^2 m_i^2, l_i^3 m_i^3, f_i h_i\}$ if $T(x_i) = \text{true}$; otherwise, if $T(x_i) = \text{false}$, we add $\bigcup_{1 \leq j \leq 3} \{r_i^j m_i^j, t_i^j g_i^j\}$.

Conversely, suppose $S \subseteq U$ is a maximal edge matching of H . Because of maximality, for each clause gadget $H(x_j)$ there exists at least one crossing edge in S incident to a vertex of $H(x_j)$. This means that there is an assignment T which satisfies all clauses of I . We now show that T is a valid assignment:

- If $H(x_i)$ complies with case (1), by maximality either $l_i^1 m_i$ or $l_i^2 m_i$ (not both) is in S , hence we set $T(x_i) = \text{true}$ (resp. $T(x_i) = \text{false}$) if $l_i^1 m_i^1 \in S$ (resp. $l_i^2 m_i$),
- If $H(x_i)$ complies with case (2), by maximality either $S_1 = \{p_i^1 p_i^3, m_i^2 r_i^2, p_i^5 p_i^7, l_i^1 m_i^1\}$ or $S_2 = \{p_i^2 p_i^3, l_i^2 m_i^2, p_i^6 p_i^7, m_i^1 r_i^1\}$ (not both) is in S , so we set $T(x_i) = \text{true}$ (resp. $T(x_i) = \text{false}$) if $S_1 \in S$ (resp. $S_2 \in S$),
- If $H(x_i)$ complies with case (3), by maximality either $S_1 = \bigcup_{1 \leq j \leq 3} \{l_i^j m_i^j\}$ or $S_2 = \bigcup_{1 \leq j \leq 3} \{r_i^j m_i^j\}$ (not both) is in S , hence we set $T(x_i) = \text{true}$ (resp. $T(x_i) = \text{false}$) if $S_1 \in S$ (resp. $S_2 \in S$).

□

5.4.2 Parameterized complexity

We now study EXT r -DCPS in the framework of parameterized complexity. For EXT r -DCPS, we can exploit structural properties of *yes*-instances and use the polynomial solvability of SIMPLE b -MATCHING to show the following.

We are dealing with the dual parameterization of EXT r -DCPS. It is sometimes more convenient to think about this problem as follows: Given is a graph $G = (V, E)$ and an edge set $\bar{U} = E \setminus U$; the question is if there exists an inclusion-wise maximal edge subset $S \subseteq E$ of G such that the partial subgraph G_S has maximum degree r and avoids \bar{U} , i.e. $S \cap \bar{U} = \emptyset$. Our parameter is $|\bar{U}|$. Assume there is an inclusion-wise maximal partial subgraph $G_{E'} = (V, E')$ of G with maximum degree r such that $E' \cap \bar{U} = \emptyset$. The next property is quite immediate.

Lemma 5.21. *The set $\{v \in V : v \text{ is incident to } \bar{U} \text{ and to } r \text{ edges in } E'\}$ is a vertex cover of $G_{\bar{U}} = (V, \bar{U})$.*

Proof. By contradiction, suppose there is an edge $xy \in \bar{U}$ such that x and y are both incident to less than r edges in E' . Then $E' \cup \{e\}$ is a new solution of r -DCPS; this is a contradiction to the maximality of E' . □

We now introduce some notations useful in the following. For a minimal vertex cover V' of $G_{\bar{U}}$, let (G', w') be the edge-weighted graph of (V, U) defined as follows: for $v \in V$, $d'(v) = 0$ if $v \notin V'$, and $d'(v) = 1$ if $v \in V'$. We define w' , the edge weight by: $w'(e) = d'(x) + d'(y)$ for $e = xy \in U$. Figure 5.10 proposes an illustration on the construction of (G', w') from an original graph G and a specified vertex cover V' of $G_{\bar{U}}$.

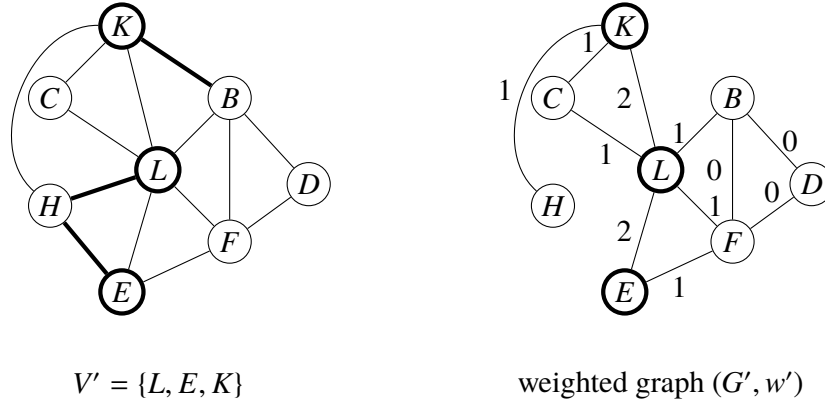


Figure 5.10: Graph $G = (V, E)$ and forbidden set $\overline{U} \subset E$ (edges are marked with bold line). A vertex cover V' of $G_{\overline{U}}$ (vertices with a bold border) is shown on the left side. On the right side, the weighted graph $G' = (V, U)$ built from G and V' .

Theorem 5.22. *There is a maximal r -DCPS set S for G such that $S \cap \overline{U} = \emptyset$ if and only if there is a vertex cover V' of $G_{\overline{U}}$ such that there exists an r -DCPS set S' for the corresponding weighted graph G' such that $w'(S') \geq |V'| \times r$.*

Proof. Let $G = (V, E)$ be a graph and let $\overline{U} \subseteq E$. Let $G_S = (V, S)$ be any maximal partial subgraph with maximum degree r of G such that $S \cap \overline{U} = \emptyset$ (if any). First observe that since S is an r -DCPS set for G which avoids \overline{U} , it is also an r -DCPS set for G' . From Lemma 5.21, we know $V' = \{v \in V : v \text{ is incident once to } \overline{U} \text{ and } r \text{ times to } S\}$ is a vertex cover of $G_{\overline{U}}$, and let $V'' \subseteq V'$ be a minimal vertex cover of $G_{\overline{U}}$. Hence, for the graph G'' with edge-weight w'' associated to the minimal vertex cover V'' of $G_{\overline{U}}$, it follows that $w''(S) \geq \sum_{v \in V''} d_{G_S}(v) = |V''| \times r$.

Conversely, assume that V' is a minimal vertex cover of $G_{\overline{U}}$ such that there exists an r -DCPS set S' for G' such that $w'(S') \geq |V'| \times r$. By the definition of the edge-weights w' , it follows that $w'(S') = \sum_{v \in V'} d_{G_{S'}}(v)$. As the subgraph $G_{S'}$ has maximum degree r , the weight $w'(S') \geq |V'| \times r$ is only possible if $d_{G_{S'}}(v) = r$ for all $v \in V'$. Greedily extending S' to an r -DCPS set for the original graph G hence gives a solution for EXT r -DCPS on (G, U) ; observe that the edges in \overline{U} can not be chosen by the greedy procedure, as each edge in \overline{U} is already incident to at least one vertex in V' which already has degree r . \square

Using the characterization given in Theorem 5.22, the next result is straightforward.

Theorem 5.23. *EXT r -DCPS, with dual parameter, is in FPT.*

Proof. Given an instance (G, U) of EXT r -DCPS, consider an algorithm that lists all minimal vertex covers V' of $G_{\overline{U}}$ and checks if the optimal value of MAX r -DCPS on (G', w') is at least $|V'| \times r$. According to Theorem 5.22, this procedure suffices to decide if there exists an r -DCPS set S with $S \cap \overline{U} = \emptyset$, i.e. if (G, U) is a yes-instance for EXT r -DCPS.

The running time is dominated by the procedure that lists all minimal vertex covers as solving MAX r -DCPS can be done in polynomial time by [85]. As the number of edges in a graph is an upper bound on any minimal vertex cover of that graph, it is clear that we can enumerate all minimal vertex covers of $G_{\overline{U}}$ in time $O^*(2^{|\overline{U}|})$ by [59, 80, 82]. \square

When bounding the degree of the graphs, we can consider an even smaller parameter and obtain feasibility results.

Proposition 5.24. *For graphs with maximum degree $r + 1$, EXT r -DCPS is polynomial-time decidable when $r = 1$ and is in FPT with respect to the number of isolated edges in \overline{U} for $r \geq 2$.*

Proof. Consider the partial subgraph $G' = G_{\overline{U}} = (V, E \setminus U)$, i.e. the graph induced by the forbidden edges. Since maximum degree of G is $r + 1$, so if G' contains a triangle or a path of length at least 3, the answer is *no*. Therefore, we can suppose that G' is a collection of stars. If one leaf of a star of G' has a degree at most r in G , then this star might be an isolated edge in G' and it is exactly for one of these two endpoints (otherwise, the answer is *no*). Hence, let $P_{G'}$ be the set of the stars which are isolated edges in G' and such that both endpoints are of degree $r + 1$ in G .

- For $r \geq 2$, for the set of stars of G' with more than one edge, leaves and center are clearly determined and for each single edge of G' not in $P_{G'}$, the endpoint with degree less than r is chosen as a center. Now, for each star in $P_{G'}$ we have to determine one of the endpoints as a center and the other one as a leaf. We can now build the set L of leaves for all stars of G' . Let $E'_L = \{uv \in E : u \in L\}$ and $G'_L = (V, E'_L)$. We check for all possible labelings, if there is a label which satisfies two following conditions the answer is *yes*, else the answer is *no*.

1. for each $v \in L$, $d_{G'_L}(v) = r + 1$.
2. for each vertex $v \in V \setminus L$, $d_{G'_L}(v) \leq r$.

- For $r = 1$, we construct a new graph H by omitting all sets of vertices $\{u'_i, u_i, v_i, w_i, w'_i\}$ for the stars $[u_i, v_i, w_i]$ of G' (with center v_i) where u'_i, w'_i are neighbors of the leaves u_i, w_i (without v_i). Notice at each time u'_i, w'_i have to be disjoint from one star to another one, otherwise the answer is *no*. Now H is a collection of paths (maybe trivial) and cycles where the forbidden edges induce a matching. Remove from H all cycles and the paths where both end edges are in U . Now H is a collection of paths where at least one of end edges is forbidden. For all of these paths, start from one side and satisfy the maximality by assigning the first possible edge to a forbidden edge, if there is a path does not satisfy the maximality, the answer is *no*, else the answer is *yes*. \square

Remark 5.25. For graphs with maximum degree $r + 1$, EXT r -DCPS with $r \geq 2$ is parameterized equivalent to SAT with respect to the number of isolated edges in $E \setminus U$ and variables, respectively.

5.4.3 Price of extension

Considering EXT 1-DCPS, we need to adapt the notion of the price of extension because we have to consider subset of forbidden edges (i.e. \overline{U}); more precisely, we want to increase $|U|$ as few as possible. Hence, the optimization problem called MIN EXT 1-DCPS is defined as follows:

EXT_{min}1-DCPS

Input: A graph $G = (V, E)$ and a set of edges $U \subseteq E$.

Solution: Maximal matching S of G .

Output: Minimize $|U \cup S|$.

For EXT_{min}1-DCPS, we denote the value of an optimal solution by $opt_{EXT_{min}1-DCPS}(G, U)$. Since $opt_{EXT_{min}1-DCPS}(G, U) \leq |U|$ with equality if and only if (G, U) is a *yes*-instance of EXT 1-DCPS, we deduce that EXT_{min}1-DCPS is NP-hard. Adapted by Definition 2.4, we define the concept of *price of extension* for EXT_{max}EC as follows:

Definition 5.26. For an instance (G, U) of EXT_{min}1-DCPS, PoE(S) of a maximal matching S of G is defined by $PoE(S) := \frac{|S \cup U|}{opt_{EXT_{min}1-DCPS}(G, U)}$.

We say that EXT_{min}1-DCPS admits a polynomial ρ -PoE if for every instance (G, U) , we can compute a matching S of G in polynomial time which satisfies $PoE(S) \leq \rho$. In the particular case $U = \emptyset$, EXT_{min}1-DCPS is exactly the well known problem *minimum maximal matching* denoted by MIN-MAX MATCHING where the goal is to find the smallest maximal matching. In [73, 75], it is shown that MIN-MAX MATCHING is hard to approximate with a factor better than 2 and 1.18, assuming Unique Games Conjecture (UGC) and $P \neq NP$, respectively. We complement this bound by showing the following.

Theorem 5.27. A 2-approximation for EXT_{min}1-DCPS can be computed in polynomial time.

Proof. Let $(G = (V, E), U)$ be an instance of EXT_{min}1-DCPS. The approximate solution for EXT_{min}1-DCPS consists in outputting a maximal matching M' of G in the following way: we build first a maximum matching in G_U and then we extend it into a maximal matching of the whole graph. Let M^* be an optimal solution of EXT_{min}1-DCPS with value $opt_{EXT_{min}1-DCPS}(G, U) = |U \cup M^*|$, i.e. a maximal matching of G containing a minimum number of edges outside of U . For a subset $A \in \{U, \bar{U}\}$, and a matching M , let $M_A = A \cap M$. Now we claim that M' is a 2-approximation algorithm for EXT_{min}1-DCPS.

By the decreasing of PoE in term of U , and considering that $|M_U^*| \leq |U|$ together with $|M_U^*| \leq |M_U'|$ (M_U' is a maximum matching in G_U), we deduce:

$$PoE = \frac{|U| + |M_{\bar{U}}'|}{|U| + |M_{\bar{U}}^*|} \leq \frac{|M_U^*| + |M_{\bar{U}}'|}{|M_U^*| + |M_{\bar{U}}^*|} \leq \frac{|M'|}{|M^*|} \quad (5.1)$$

It is well known that that $|M_1| \leq 2|M_2|$ for any pair M_1, M_2 of maximal matchings. So in particular, putting this inequality with $M_1 = M'$ and $M_2 = M^*$ in inequality (5.1), gives the expected result. \square

5.5 Conclusion

In this chapter, we considered the extension variant of some edge graph problems. Given a graph $G = (V, E)$ and a subset $U \subseteq E$ as an input of an anti-hereditary (resp. a hereditary) problem

Π , it is asked, if there exists a minimal (resp. maximal) solution S such that $U \subseteq S$ (resp. $U \supseteq S$). Particularly, we considered the extension variant of the generalizations of **EDGE COVER**, **MATCHING** and **EDGE DOMINATING SET**, denoted by **EXT r -EC**, **EXT r -EDS** and **EXT r -DCPS** respectively.

We have shown all the problems are NP-complete for graphs of maximum degree $r + 2$, moreover we proved for the particular case $r = 1$, all of them are NP-complete for sub-cubic planar bipartite instances. We also studied the parameterized complexity of these problems, with parameter $|U|$ or $|\bar{U}|$. We proposed some positive parameterized results, **EXT r -EC** and **EXT r -EDS** (resp. **EXT r -DCPS**) with respect to the standard parameter (resp. the dual parameter) are in FPT. Table 5.1 summarizes the complexity and parameterized results obtained for the these problems.

| | NP-completeness | | Parameterized Complexity |
|--------------------------------|---|-----------------------------------|--------------------------|
| | $r \geq 2$ | $r = 1$ | |
| EXT r-EC | bipartite graphs with $\Delta = r + 2$ | sub-cubic planar bipartite graphs | FPT by $ U $ |
| EXT r-EDS | planar bipartite graphs with $\Delta = r + 2$ | sub-cubic planar bipartite graphs | W[1]-hard by $ U $ |
| EXT r-DCPS | bipartite graphs with $\Delta = r + 1$ | sub-cubic planar bipartite graphs | FPT by $ \bar{U} $ |

Table 5.1: The computational complexity and parameterized results.

Furthermore, we have discussed the *price of extension* (PoE), measuring the distance of U to the closest set that can be extended, which results in natural optimization problems related to extension problems for which we discuss polynomial-time approximability.

Our results should be useful in particular to the (input-sensitive) enumeration algorithms community that has so far not put that much attention on edge graph problems; we are only aware of [95] in this direction. Conversely, output-sensitive enumeration algorithms, say, for matchings have been around of more than twenty years [157]. Some thoughts on edge cover enumeration can be found in [160].

General Framework of Extension Problems

Contents

| | |
|--|------------|
| 6.1 A General framework of extension problems | 168 |
| 6.2 Some partial orders | 170 |
| 6.3 Some new extension problems | 171 |
| 6.4 Conclusion | 176 |

The question "if a given partial solution to a problem can be extended" occurs in many algorithmic approaches for optimization problems. In Chapters 4 and 5, we studied some vertex and edge graph optimization problems in this framework. However, there exists a great number of NP optimization problems like string problems, logic problems and graph coloring problems, etc, whose extension variants are interesting. Actually, many of these problems have been considered in the framework of extension problems [23, 109, 84, 7, 8, 113, 139, 109, 146, 14, 159, 138, 23] however, there is a crucial difference with our approach embodied by the necessity of the minimality condition in our framework. Our main intention in this chapter is to propose a general framework of such extension problems. This framework is based on a partial order approach, reminiscent of what has been endeavored for *max-min* or *min-max* framework in [136]. In essence, we consider optimisation problems in NPO with an additionally specified set of partial solutions which we call *pre-solutions* (including the set of solutions) and a partial order on those. This partial order \leq reflects not only the notion of *extension* but also of *minimality* as follows. For a pre-solution U and a solution S , S *extends* U if $U \leq S$. A solution S is *minimal* if there exists no solution $S' \neq S$ with $S' \leq S$. The resulting extension problem is defined as the task to decide for a given pre-solution U , if there exists a minimal solution S which extends U .

The chapter is organized as follows. We introduce a general framework for extension problems based on partial order in Section 6.1. In Section 6.2, we will present some interesting partial orders for NP optimization problem. We finish this chapter, by proposing some new hardness results for the extension version for problems like BIN PACKING, FEEDBACK VERTEX SET and FEEDBACK EDGE SET.

6.1 A General framework of extension problems

We begin here by giving a general definition of *partial order*.

Definition 6.1. A relation \leq is a partial order on a set X if it has:

- for each $S, S' \in X$, $S \leq S'$ and $S' \leq S$ implies $S = S'$,
- for each $S, S', S'' \in X$, $S \leq S'$ and $S' \leq S''$ implies $S \leq S''$.

In order to formally define our concept of minimal extension, we define what we call *monotone problems* which can be thought as problems in NPO with the addition of a set of pre-solutions (which includes the set of feasible solutions) together with a partial ordering on this new set. Formally we define such problems as 5-tuples $\Pi = (I, \text{presol}, F, \leq, m)$ (where I, F, m with an additional $\text{opt} \in \{\min, \max\}$ yields an NPO problem) is defined by:

- I is the set of instances, recognizable in polynomial time.
- For $x \in I$, $\text{presol}(x)$ is the set of *pre-solutions* and, in a reasonable representation of instances and pre-solutions, the length of the encoding of any $y \in \text{presol}(x)$ is polynomially bounded in the length of the encoding of x .
- For $x \in I$, $F(x)$ is the *set of feasible solutions*, which is a subset of $\text{presol}(x)$
- There exists an algorithm which, given (x, y) , decides in polynomial time if $y \in \text{presol}(x)$; similarly there is an algorithm which decides in polynomial time if $y \in F(x)$. This is important for the purposes of showing membership in NP or NPO.
- For $x \in I$, \leq_x is a partial ordering on $\text{presol}(x)$ and there exists an algorithm that, given an instance x and $y, z \in \text{presol}(x)$, can decide in polynomial time if $y \leq_x z$.
- For each $x \in I$, the set of solutions $F(x)$ is upward closed with respect to the partial ordering \leq_x , i.e. $U \in F(x)$ implies $U' \in F(x)$ for all $U, U' \in \text{presol}(x)$ with $U \leq_x U'$.
- m is a polynomial-time computable function which maps pairs (x, y) with $x \in I$ and $y \in \text{presol}(x)$ to non-negative rational numbers; $m(x, y)$ is the *value of y* .
- For $x \in I$, $m(x, \cdot)$ is *monotone* with respect to \leq_x , meaning that the property $y \leq_x z$ for some $y, z \in \text{presol}(x)$ either always implies $m(x, y) \leq m(x, z)$ or $m(x, y) \geq m(x, z)$.

The requirement that the set of solutions is upward closed with respect to the partial ordering relates to *independence systems*, see [151].

Definition 6.2. Let $\Pi = (I, \text{presol}, F, \leq, m)$ be a monotone problem. For an instance $x \in I$, the set of minimal feasible solutions of x denoted by $\mu(x, \leq_x)$ is:

$$\mu(x, \leq_x) = \{S \in F(x) : ((S' \leq_x S) \wedge (S' \in F(x))) \rightarrow S' = S\}.$$

Further, given $U \in \text{presol}(x)$, we define $\text{ext}(x, U) = \{U' \in \mu(x, \leq_x) : U \leq_x U'\}$ to be the set of *extensions* of U . Sometimes, $\text{ext}(x, U) = \emptyset$, which makes the question of the existence of such extensions interesting. Hence, finally, the *extension problem* for Π , written $\text{EXT } \Pi$, is defined as follows:

EXT Π

Input: An instance $x \in I$, a pre-solution $U \in \text{presol}(x)$ and a partial order \leq_x .

Question: Does x have a \leq_x -minimal solution S , such that $U \leq_x S$?

Although we strongly linked the definition of monotone problems to NPO, the corresponding extension problems do not generally belong to NP (in contrast to the canonical decision problems associated to NPO problems). Consider the monotone problem $\Pi_\tau = (I, \text{presol}, F, \leq, m)$ with:

- $I = \{x : x \text{ is a Boolean formula}\}$.
- For a formula $x \in I$ on n variables, $\text{presol}(x) = F(x) = \{\phi \mid \phi : \{1, \dots, n\} \rightarrow \{0, 1\}\}$.
- For $\phi, \psi \in \text{presol}(x)$, $\phi \leq_x \psi$ if either $\phi = \psi$, or assigning variables according to ψ satisfies x while an assignment according to ϕ does not.
- $m \equiv 1$ (plays no role for the extension problem)

The associated extension problem $\text{EXT } \Pi_\tau$ corresponds to the co-NP-complete TAUTOLOGY PROBLEM in the following way: Given a Boolean formula x which, w.l.o.g., is satisfied by the all-ones assignment $\psi_1 \equiv 1$, it follows that (x, ψ_1) is a *yes*-instance for $\text{EXT } \Pi_\tau$ if and only if x is a tautology, as ψ_1 is in $\mu(x, \leq_x)$ if and only if there does not exist some $\psi_1 \neq \phi \in F(x)$ with $\phi \leq_x \psi_1$, so, by definition of the partial ordering, an assignment ϕ which does not satisfy x . Consequently $\text{EXT } \Pi_\tau$ is not in NP, unless co-NP = NP.

In Chapter 4 and Chapter 5, we have mentioned some well-known monotone graph problems, for which I is the set of undirected graphs, denoting instances by $x = G(V, E)$, and $m(x, U) = |U|$ for all $U \in \text{presol}(x)$. In the following, we remind all of these problems:

- VERTEX COVER (VC): $\leq_x = \subseteq$, $\text{presol}(x) = 2^V$, $S \in F(x)$ if and only if each $e \in E$ is incident to at least one $v \in S$;
- EDGE COVER (EC): $\leq_x = \subseteq$, $\text{presol}(x) = 2^E$, $S \in F(x)$ if and only if each $v \in V$ is incident to at least one $e \in S$;
- DOMINATING SET (DS): $\leq_x = \subseteq$, $\text{presol}(x) = 2^V$, $S \in F(x)$ if and only if $N[S] = V$;
- EDGE DOMINATING SET (EDS): $\leq_x = \subseteq$, $\text{presol}(x) = 2^E$, $S \in F(x)$ if and only if each edge belongs to S or is adjacent to some $e \in S$.
- CONNECTED VERTEX COVER (CVC): $\leq_x = \subseteq$, $\text{presol}(x) = 2^V$, $S \in F(x)$ if and only if each $e \in E$ is incident to at least one $v \in V$ and $G[S]$ is connected.

- INDEPENDENT SET (IS): $\leq_x = \supseteq$, $presol(x) = 2^V$, $S \in F(x)$ if and only if $G[S]$ contains no edges;
- EDGE MATCHING (EM): $\leq_x = \supseteq$, $presol(x) = 2^E$, $S \in F(x)$ if and only if none of the vertices in V is incident to more than one edge in S ;
- NON SEPARATING INDEPENDENT SET (NSIS): $\leq_x = \supseteq$, $presol(x) = 2^V$, $S \in F(x)$ if and only if S is an independent set and it is not a cut-set of G .

We hence arrived at problems like EXT VC (resp. EXT IS), where the instance is specified by a graph $G = (V, E)$ and a vertex set U , and the question is if there is some *minimal* vertex cover $S \supseteq U$ (resp. some *maximal* independent set $I \subseteq U$), or for the problems like EXT EC (resp. EXT EM), where the instance is specified by a graph $G = (V, E)$ and an edge set U , and the question is if there is some *minimal* edge cover $S \supseteq U$ (resp. some *maximal* matching $M \subseteq U$). So far, we only considered problems with $\leq = \subseteq$ or $\leq = \supseteq$. In [136], a rich source of instance ordering for different problems were proposed. Some of the most interesting is introduced in the following.

6.2 Some partial orders

Some partial orders related to *maximum-minimal* and *minimum-maximal* optimization problems were proposed by Manlove in [136]: He applied the partial order *set inclusion* on 12 distinct covering and independent problems. He also proposed many other partial orders on different types of optimization problems, some of them are represented in the following:

Coloring problems. two partial orders on the set of all proper coloring for a given graph G :

- The partial order "*partition merge*" gives rise to ACHROMATIC NUMBER problem which is finding a proper coloring of G with maximum number of colors such that every pairs of colors appears at the endpoints of some edge of G .
- The partial order "*partition redistribution*" gives rise to BACHROMATIC NUMBER problem which is finding a proper coloring with maximum number of colors such that for each color i , there is a distinguished vertex of color i which has the vertices of every colors in his neighborhood.

Graph problems with another partial order. the partial order $(k - 1, k)$ -*replacement* is applied to the set of all independent set for a given graph $G = (V, E)$.

- An independent set $S \subseteq V$ is maximal respect to the partial order $(k - 1, k)$ -*replacement* if removing of any $r - 1$ vertices from S , together with adding of any r vertices from $V \setminus S$ (for any $r \leq k$), leads to a non-independent set.

String problems. the partial orders "*subsequence*", "*substring*", "*supersequence*" and "*superstring*" concerning source string problems.

- For a given set of strings S , s is a maximal common subsequence (resp. substring) of S if (i) s is a subsequence (resp. substring) of each string $t \in S$, (ii) there does not exist a common string s' , supersequence (resp. superstring) of s . The definition of minimality is analogous.

Data storage problems. two partial orders concerning a source well-known optimization problem BIN PACKING. The input of BIN PACKING consists of a set $X = \{x_1, \dots, x_n\}$ of items, a weight function w that associates non-negative integers to items and a positive integer B which shows the capacity of every bin. A feasible solution is a partition π of X such that, for each set $Y \in \pi$, $\sum_{y \in Y} w(y) \leq B$. The traditional aim is to find a feasible π such that $|\pi|$ is minimized.

- given an instance of minimum BIN PACKING, a partition π is minimal respect to the partial order "*partition merge*", if it is not possible to merge the content of any two bins of π into a single bin, without exceeding the bin capacity.
- a partition π is minimal respect to the partial order "*partition redistribution*", if it is not possible to redistribute the content of a bin of π amongst the remaining bins, without exceeding the bin capacity.

Logic problems. two partial orders "*truth assignment-1*" and "*truth assignment-2*" concerning source logic problem SAT.

- for an instance $I = (\mathcal{C}, \mathcal{X})$ of SAT, let $\leq_I = "$ *truth assignment-1* $"$, then for two truth assignments f, g of I , $f \leq_I g$ if g satisfies all of clauses of \mathcal{C} satisfied by f , plus at least one more, and f, g give the same assignment for all the variables which appear in the true clauses of f .
- the partial order "*truth assignment-2*" has the same definition of "*truth assignment-1*", but not requiring the two truth assignments f, g to have the same assignments for all the variables which appear in the true clauses of f .

6.3 Some new extension problems

In this section, we introduce some new extension problems which didn't fit in Chapters 4 and 5 and proposed some hardness results for each of them.

A *feedback vertex set* in a graph $G = (V, E)$ is a subset S of vertices such that $G[V \setminus S]$ is acyclic. Two variants of this problem, depending on G is directed or undirected, have been introduced in the literature. In [88], it is shown that it is NP-hard to find a feedback vertex set or feedback arc set of minimum size for a given graph. In the following, we formally define the extension variant of feedback vertex set and directed feedback vertex set problems denoted by EXT FVS and EXT DFVS respectively.

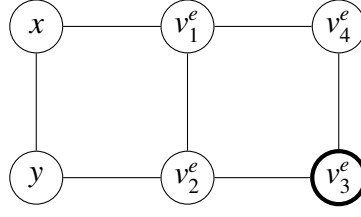


Figure 6.1: Gadget $H(e)$ for $e = xy$. The vertex v_3^e is in U'

EXT FVS

Input: A graph $G = (V, E)$ together with a pre-solution $U \subseteq V$.

Question: Does G have a minimal feedback vertex set S with $S \supseteq U$?

EXT DFVS

Input: A directed graph $G = (V, A)$ together with a pre-solution $U \subseteq V$.

Question: Does G have a minimal feedback vertex set S with $S \supseteq U$?

Here, we show that the extension variant of both problems are NP-hard in graphs of bounded degree.

Theorem 6.3. *EXT FVS is NP-complete in bipartite graphs of maximum degree 6.*

Proof. We make a simple reduction from EXT VC to EXT FVS. Let (G, U) be an instance of EXT VC where $G = (V, E)$ is a bipartite graph of maximum degree 3 and $U \subseteq V$ is a forced set, we make a new graph $G' = (V', E')$ by adding a gadget $H(e) = (V_e, E_e)$ containing 4 new vertices $v_1^e, v_2^e, v_3^e, v_4^e$ and 6 new edges $xv_1^e, yv_2^e, v_1^ev_2^e, v_2^ev_3^e, v_3^ev_4^e, v_4^ev_1^e$ for each $e = xy \in E$ to G . Formally:

- $V' = V \cup \{v_1^e, v_2^e, v_3^e, v_4^e : e \in E\}$,
- $E' = E \cup \{xv_1^e, yv_2^e, v_1^ev_2^e, v_2^ev_3^e, v_3^ev_4^e, v_4^ev_1^e : e = xy \in E\}$.

An illustration of gadget $H(e)$ for $e = xy$ is described in Figure 6.1. Obviously, G' is a bipartite graph of maximum degree 6 and can be made easily from G in polynomial time. Now, we claim that (G, U) is a *yes*-instance of EXT VC if and only if (G', U') with $U' = U \cup \{v_3^e : e \in E\}$ is a *yes*-instance of EXT FVS.

Suppose S is a minimal vertex cover containing U , then easily one can see that $S' = S \cup \{v_3^e : e \in E\}$ is a minimal feedback vertex set of G' containing U' . Conversely, suppose that S be a minimal feedback vertex set containing U' , by minimality, for each $e \in E$, $S \cap V_e = v_3^e$, so to remove all the cycles of the graph, at least one of x, y is in S . \square

Theorem 6.4. *EXT DFVS is NP-complete in bipartite graphs of maximum in-degree and out-degree 3.*

Proof. For EXT DFVS, we use a reduction from EXT VC adapted by Karp's reduction [118]. Let (G, U) with $G = (V, E)$ and $U \subseteq V$ be an instance of EXT VC, we transform G into a digraph $G' = (V, A)$, with the same vertices as G and replacing two arcs (u, v) and (v, u) to A for each edge $uv \in E$. Note that G' is bipartite of maximum in-degree and out-degree 3 if G is bipartite of maximum degree 3. For each edge $uv \in E$, at least one of vertices u or v is in any minimal vertex cover. Further, since for each edge in G , we make a directed cycle in G' , so for each edge $uv \in E$ at least one of vertices u or v is in any minimal directed feedback vertex set of G' . Hence, it is easy to see that (G, U) is a *yes*-instance of EXT VC if and only if (G', U) is a *yes*-instance of EXT DFVS. \square

A *feedback edge set* in a graph $G = (V, E)$ is a subset S of edges such that the deletion of $G[V \setminus S]$ is acyclic. Two variations of the problem, depending on G is directed or undirected have been considered in the literature. In [90], it is shown that MIN DIRECTED FEEDBACK EDGE SET (MIN DFES for short) is NP-hard in graphs of maximum in-degree and out-degree 3, while MIN FEEDBACK EDGE SET (MIN FES for short) in polynomial solvable even for edge weighted graph. Actually the former problem is equivalent to find a spanning tree of maximum size for a given edge weighted graph. The extension variant of feedback edge set and directed feedback edge set, denoted by EXT FES and EXT DFES are as follows:

EXT FES

Input: A graph $G = (V, E)$ together with a pre-solution $U \subseteq E$.

Question: Does G have a minimal feedback edge set S with $S \supseteq U$?

EXT DFES

Input: A directed graph $G = (V, A)$ together with a pre-solution $U \subseteq A$.

Question: Does G have a minimal feedback edge set S with $S \supseteq U$?

Theorem 6.5. *EXT DFES is NP-complete in bipartite graphs.*

Proof. We propose a reduction from EXT VC adapted by the reduction given in [118]. Let (G, U) with $G = (V, E)$ and $U \subseteq V$ be an instance of EXT VC, we transform G into a digraph $G_A = (V_A, A)$ by duplicating each vertex of G and by adding a set of circuit of size 2. More precisely, $V_A = V \cup \{v' : v \in V\}$ and the arc set $A = A' \cup A''$ where $A' = \{(u, u'), (v, v')\}$ and $A'' = \{(u, v'), (v', u), (v, u'), (u', v) : e = uv \in E\}$. An illustration of this reduction for an edge $e = uv$ is depicted in Figure 6.2. Finally, we define a forced set of arcs $U_A = \{(u, v'), (v, u') : e = uv \in E\} \cup \{(u, u') : u \in U\}$. Obviously, G_A is bipartite of maximum in-degree and out-degree 4 if G is bipartite of maximum degree 3. Now, we claim that (G, U) is a *yes*-instance of EXT VC if and only if (G_A, U_A) is a *yes*-instance of EXT DFES.

Suppose S is a minimal vertex cover of G containing U . By setting $S' = \{uu' : u \in S\} \cup U_A$, we obtain a minimal directed feedback edge set of G_A . Conversely, if S is a minimal directed feedback edge set of G_A containing U_A . Note that any minimal feedback edge set of G_A that includes U_A cannot contain any other edge from A'' . Hence, easily by setting $S' = \{u \in V : uu' \in S\}$ we have a minimal connected vertex cover of G which contains U . \square

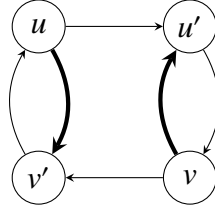


Figure 6.2: Example of construction of G_A if G is a single edge $e = uv$. Arcs drawn in bold are in U_A .

Theorem 6.6. *EXT FES is polynomial solvable.*

Proof. Note that a feedback edge set S of a given connected graph $G = (V, E)$ is minimal if the partial graph $(V, E \setminus S)$ is a tree. Hence, for a given connected graph $G = (V, E)$ and a subset of forced edges $U \subseteq E$, there is a minimal feedback edge set of G containing U if the partial graph $(V, E \setminus U)$ is a connected component. Generally, let (G, U) be an instance of EXT FES where $G = (V, E)$ is a simple graph and $U \subseteq E$, then the answer is yes if and only if the number of connected components of partial graph $(V, E \setminus U)$ is equals to the number of connected components of G . \square

So far, only the partial ordering *set-inclusion* is used in our examples. However, the reader can refer to [136] as a rich source of further instance orderings. Here, we mention one other example.

Bin Packing. Here, we make use of the well-known partition ordering. We already introduced a definition of a well known optimization problem BIN PACKING together with two different partial orderings. Here, we propose an equivalent definition for the problem and define the extension variant of the problem. The input of BIN PACKING consists of a set $X = \{x_1, \dots, x_n\}$ of items and a weight function w that associates rational numbers $w(x_i) \in (0, 1)$ to items. A feasible solution is a partition π of X such that, for each set $Y \in \pi$, $\sum_{y \in Y} w(y) \leq 1$. The traditional aim is to find a feasible π such that $|\pi|$ is minimized.

Now, $presol(X)$ collects all partitions of X . For two partitions π_1, π_2 of X , we write $\pi_1 \leq_X \pi_2$ if π_2 is a refinement of π_1 , i.e. π_2 can be obtained from π_1 by splitting up its sets into a larger number of smaller sets. Hence, $\{X\}$ is the smallest partition with respect to \leq_X . As a partition π is a set, we can measure its size by its cardinality. Clearly, the set of solutions is upward closed. Now, a solution is minimal if merging any two of its sets into a single set yields a partition π such that there is some $Y \in \pi$ with $w(Y) := \sum_{y \in Y} w(y) > 1$. In the extension variant, we are given a partition π_U of X (together with X and w) and ask if there is any minimal feasible partition π'_U with $\pi_U \leq_X \pi'_U$. This describes the problem EXT BP. One could think of encoding knowledge about which items should not be put together in one bin within the given partition π_U . To be clear, let us to give an example.

Example 6.7. Let $X = \{x_1, x_2, x_3, x_4, x_5\}$, a set of items where $w(x_1) = 0.2, w(x_2) = 0.4, w(x_3) = 0.5, w(x_4) = 0.7, w(x_5) = 0.9$ be an instance of bin packing problem. Consider two pre-solution $\pi_{U_1} = \{\{x_1, x_2, x_3\}, \{x_4, x_5\}\}$ and $\pi_{U_2} = \{\{x_1, x_5\}, \{x_2, x_3, x_4\}\}$.

- (X, π_{U_1}) is a *yes*-instance of EXT BP, because we can make a partition $\pi = \{\{x_1, x_2\}, \{x_3\}, \{x_4\}, \{x_5\}\}$ which is $\pi_{U_1} \leq_X \pi$ and is minimal.
- (X, π_{U_2}) is a *no*-instance of EXT BP, because to have a feasible solution for bin packing, we have to split each of $\{x_1, x_5\}$ and $\{x_2, x_3, x_4\}$ into some new sets. the new partition is not minimal, as we can join x_1 and x_4 .

In the following, we show that the EXT BP is an NP-hard problem. In this regard, we make a reduction from 3-PARTITION which asks for a given multiset $S = \{s_1, \dots, s_{3m}\}$ of integers and $b \in \mathbb{N}$ if S can be partitioned into m triples such that the sum of each subset equals b .

Theorem 6.8. *EXT BP is NP-hard, even if the pre-solution π_U contains only two sets.*

Proof. The proof consists of a reduction from 3-PARTITION which is defined as follows: given a multiset $S = \{s_1, \dots, s_{3m}\}$ of positive integers and a positive integer b as input, decide if S can be partitioned into m triples S_1, \dots, S_m such that the sum of each subset equals b . 3-PARTITION is NP-complete even if each integer satisfies $b/4 < s_i < b/2$; see [88].

Let $(S = \{s_1, \dots, s_{3m}\}, b)$ be the input of 3-PARTITION, where $b/4 < s_i < b/2$ for each $1 \leq i \leq 3m$. We build a set $X = \{x_0, x_1, \dots, x_{3m}\}$ of items and a weight function w where $w(x_0) = \frac{1}{m+1}$ and $w(x_i) = \frac{s_i}{b}$ for each $1 \leq i \leq 3m$ and set $\pi_U = \{\{x_0\}, \{x_1, \dots, x_{3m}\}\}$ as a partial partition of X . We claim that (S, b) is a *yes*-instance of 3-PARTITION if and only if (X, π_U) is a *yes*-instance of EXT BP.

Suppose first that S can be partitioned into m triples S_1, \dots, S_m where $\sum_{s_j \in S_i} s_j = b$ for each $S_i \in S$. We build a set $X_i = \{x_j : 1 \leq j \leq 3m, s_j \in S_i\}$, $1 \leq i \leq m$. Considering π_U , $\pi'_U = \{\{x_0\}, X_1, \dots, X_m\}$ is a feasible partition and since for each $S_i \in S$, $\sum_{s_j \in S_i} s_j = b$, we have $w(X_i) = 1$ for each $X_i \in \pi'_U$. Hence π'_U is not the refinement of any other feasible partition for (S, b) , as especially x_0 cannot be added to any subset $X_i \in \pi'_U$. Since π'_U is obviously a refinement of π_U , π'_U is a minimal feasible partition with $\pi_U \leq_X \pi'_U$.

Conversely, assume that π'_U is a minimal partition of X as a refinement of π_U . As the set $\{x_0\}$ in the partition π_U can not be split up further, it follows that the extension π'_U is of the form $\{\{x_0\}, X_1, \dots, X_k\}$. By using the minimality of π'_U , it follows especially that $\sum_{x_l \in X_i} w(x_l) + w(x_0) > 1$ for all $i \in \{1, \dots, k\}$, as otherwise $\pi''_U = \{X_1, \dots, X_{i-1}, X_i \cup \{x_0\}, X_{i+1}, \dots, X_k\}$ would be a feasible partition of X with $\pi''_U \leq_X \pi'_U$. We claim that $k = m$. As $k < m$ is not possible, assume that $k > m$. Since $\sum_{i=1}^{3m} w(x_i) = \frac{1}{b} \sum_{i=1}^{3m} s_i = m$, this means that $\sum_{x_l \in X_j} x_l w(X_j) \leq \frac{m}{k}$ for at least one $j \in \{1, \dots, k\}$, which contradicts $\sum_{x_l \in X_j} w(x_l) + w(x_0) > 1$ by the definition of x_0 . Consider the collection of the sets $S_i = \{s_j : 1 \leq j \leq 3m, x_j \in X_i\}$, $1 \leq i \leq m$ as a partition for S . By feasibility of π'_U , it follows that $w(X_i) \leq 1$, which means $\sum_{s_l \in S_i} s_l \leq b$ and $k = m$ implies that indeed $\sum_{s_l \in S_i} s_l = b$ for each $i \in \{1, \dots, m\}$. The requirement $b/4 < s_i < b/2$ for each $1 \leq i \leq 3m$ then implies that the size of each X_i equals 3, which overall means that S_1, \dots, S_m is a solution for 3-PARTITION on (S, b) . \square

6.4 Conclusion

In this chapter, we introduced a general framework to model extension for monotone problems with the attempt to highlight the unified structure of such types of problems that seem to appear in many different scenarios. Admittedly, our framework does not cover all problems of this flavor. Quite similar problems have for example been considered in the area of graph coloring, under the name of *pre-coloring extension*, which contains the completion of partial Latin squares as a special case [23, 54, 138]. However, there is a crucial difference with our approach: while with our problems, the minimality condition on the permissible extensions is essential for all our considerations, they become at best uninteresting for pre-coloring extension problems, although it is not difficult to define partial orderings on pre-colorings so that the set of proper colorings is upward closed as required in our setting. It would be interesting to consider such kinds of extension problems in a wider framework to have general results, meta-theorems, etc.

Conclusion and Future Work

In this manuscript, we have presented a general study of a special type of graph optimization problems known as *extension problems*. Generally, an instance of an extension graph problem in our framework consists of a graph $G = (V, E)$ together with a pre-solution U which is a subset of edges or vertices of G , depending on the problem, and the goal is to *extend* U to a feasible solution. More precisely, we study the framework of extension problems on hereditary and anti-hereditary graph optimization problems, in which, the goal is to extend the pre-solution to an inclusion-wise *maximal* and *minimal* solution, respectively. For an anti-hereditary (resp. a hereditary) extension problem, the pre-solution U is called *forced set* (resp. *co-forbidden set*) and any feasible solution S is extended of U if $S \supseteq U$ (resp. $S \subseteq U$).

We have studied different types of optimization/decision problems in the framework of extension problems on a large number of hereditary and anti-hereditary graph problems. A natural optimization problem for an anti-hereditary (resp. a hereditary) graph problem Π and for an instance (G, U) , asks a minimal solution containing U (resp. a maximal solution is included in U) while optimizing the objective function. For instance, for an anti-hereditary problem like VERTEX COVER, the input consists of a graph $G = (V, E)$ and the forced set $U \subseteq V$, and the goal is to find a minimal vertex cover S with minimum size such that $S \supseteq U$, while for a hereditary problem like EDGE MATCHING, the input consists of a graph $G = (V, E)$ and a co-forbidden set $U \subseteq E$ and the goal is to find a maximal matching included in U with maximum cardinality. In Chapter 3, we studied this optimization problem for WEIGHTED EDGE COVER problem and have considered both minimization and maximization variants of the problem on complete graphs.

Sometimes, for an instance (G, U) of a graph extension problem, there is no feasible solution, which makes an interesting decision problem, denoted by EXT Π in this manuscript. For an anti-hereditary (resp. a hereditary) problem Π , the decision problem EXT Π for an instance (G, U) asks does G have a minimal (resp. a maximal) solution S such that extended of U , i.e. $S \supseteq U$ (resp. $S \subseteq U$). Concerning the former decision problem, we have studied a large number of vertex and edge graph problems respectively.

We have investigated the decision problem EXT Π for vertex graph problems like VERTEX COVER, INDEPENDENT SET, CONNECTED VERTEX COVER, NON SEPARATING INDEPENDENT SET and

DOMINATING SET. We proved that the mentioned problems are NP-complete in very restricted scenarios, be it degree or topological restrictions (bipartite, planar or chordal graphs), while we have found many graph classes where the extension problems remains solvable in polynomial-time. We also have studied the parameterized complexity of these problems, with respect to parameter $|U|$ as well as $|\overline{U}|$.

Moreover, we have studied the decision problem EXT Π for some edge graph problems like EDGE COVER, EDGE DOMINATING SET and EDGE MATCHING and the generalized version of them. We have proposed some hardness proofs for general input graphs and also when the input graphs are restricted by degree or typologies. We also have shown some positive and negative results in parameterized complexity point of view.

Considering the possibility that some pre-solutions U might not be extensible to any minimal or maximal solution, one might ask how far is U from an extensible set. We have tried to discuss how many elements of U have to be deleted (resp. added) for an anti-hereditary (resp. a hereditary) problem Π , in order to arrive at a *yes*-instance of EXT Π . Concerning this, we have defined a new notion *price of extension* (PoE for short), in an attempt to understand what effect the additional pre-solution constraint has on the possibility of finding minimal or maximal solutions. In this regard, for an anti-hereditary and a hereditary problem Π , we have introduced two new optimization problems denoted by EXT_{min} Π and EXT_{max} Π respectively, and have studied these optimization problems for all the aforementioned vertex and edge graph problems from an (in-)approximability point of view. We also studied some graph problems in the framework of MAX-MIN and MIN-MAX optimization problems as a particular case of EXT_{min} Π (resp. EXT_{max} Π) for an instance (x, U) occurs when the pre-solution U is "empty-set" (resp. "whole-set").

Moreover, we have introduced a general partial-order based framework for extension problems with the attempt to highlight a unified structure of such types of problems which might be appeared in different scenarios. Informally, in the extension version of an optimization problem Π , an input of the corresponding extension problem consists of an instance of Π with an additionally pre-solution and a partial order, and the goal is to extend the pre-solution to a *minimal* solution with respect to the partial order. A solution S is *minimal*, if there exists no solution $S' \neq S$ with $S' \leq S$. We also have introduced an extension variant of BIN PACKING with a new partial-order *partition refinement* and proposed a hardness proof for the decision problem EXT BP.

As a future work, it would be a good idea to study further optimization problems that could be related to our extension problems, for instance, given an instance (G, U, k) of an anti-hereditary or a hereditary problem Π , is it possible to add at most k edges to the graph such that (G, U) becomes a *yes*-instance of EXT Π ? Such problems would be defined according to the general idea of graph editing problems studied quite extensively in recent years. These problems are particularly interesting in graph classes where EXT Π is solvable in polynomial time.

We have proposed a partial order based on general framework for extension problems, however, we have studied only hereditary and anti-hereditary graph problems when the partial-order is *set inclusion* or *set exclusion*. There are some potentially good problems like *string problems* or *logic problems* where can be considered with different partial-orders in our framework. Certainly, our framework does not cover all problems of this flavor. For example, in the area of graph coloring, a

problem under the name *pre-coloring extension* has been considered in the framework of extension problems with different approach where the minimality condition become meaningless. Although it is pretty straightforward to define partial orderings on pre-colorings so that the set of proper colorings is upward closed as required in our setting. It would be interesting to study such forms of extension problems also in a wider framework to have general results, meta-theorems, etc.

Bibliography

- [1] J. Abello, M. R. Fellows, and J. Stillwell. On the complexity and combinatorics of covering finite complexes. *Australasian J. Combinatorics*, 4:103–112, 1991.
- [2] H. AbouEisha, S. Hussain, V. V. Lozin, J. Monnot, B. Ries, and V. Zamaraev. A boundary property for upper domination. In V. Mäkinen, S. J. Puglisi, and L. Salmela, editors, *Proc. of IWOCA 2016*, volume 9843 of *LNCS*, pages 229–240. Springer, 2016.
- [3] D. Achlioptas. The complexity of g -free colourability. *Discrete Mathematics*, 165:21–30, 1997.
- [4] P. Alimonti and V. Kann. Some APX-completeness results for cubic graphs. *Theor. Comput. Sci.*, 237(1-2):123–134, 2000.
- [5] P. Alimonti and V. Kann. Some apx-completeness results for cubic graphs. *Theoretical Computer Science*, 237(1-2):123–134, 2000.
- [6] P. Angelini, G. D. Battista, F. Frati, V. Jelínek, J. Kratochvíl, M. Patrignani, and I. Rutter. Testing planarity of partially embedded graphs. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 202–221. SIAM, 2010.
- [7] S. Anily, J. Bramel, and A. Hertz. A $5/3$ -approximation algorithm for the clustered traveling salesman tour and path problems. *Oper. Res. Lett.*, 24(1-2):29–35, 1999.
- [8] E. M. Arkin, R. Hassin, and L. Klein. Restricted delivery problems on a network. *Networks*, 29(4):205–216, 1997.
- [9] S. Arnborg, D. G. Corneil, and A. Proskurowski. Complexity of finding embeddings in ak -tree. *SIAM Journal on Algebraic Discrete Methods*, 8(2):277–284, 1987.
- [10] S. Arora, D. Karger, and M. Karpinski. Polynomial time approximation schemes for dense instances of np-hard problems. *Journal of computer and system sciences*, 58(1):193–210, 1999.

-
- [11] S. Athanassopoulos, I. Caragiannis, C. Kaklamanis, and M. Kyropoulou. An improved approximation bound for spanning star forest and color saving. In R. Královic and D. Niwinski, editors, *Proc. of 34th MFCS*, volume 5734 of *LNCS*, pages 90–101. Springer, 2009.
- [12] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and approximation: Combinatorial optimization problems and their approximability properties*. Springer Science & Business Media, 2012.
- [13] G. Ausiello, P. Crescenzi, and M. Protasi. Approximate solution of np optimization problems. *Theoretical Computer Science*, 150(1):1–55, 1995.
- [14] M. Axenovich, J. P. Hutchinson, and M. A. Lastrina. List precoloring extension in planar graphs. *Discrete Mathematics*, 311(12):1046–1056, 2011.
- [15] R. Bar-Yehuda and S. Even. On approximating a vertex cover for planar graphs. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 303–309. ACM, 1982.
- [16] C. Bazgan, L. Brankovic, K. Casel, and H. Fernau. On the complexity landscape of the domination chain. In *Algorithms and Discrete Applied Mathematics - Second International Conference, CALDAM*, pages 61–72, 2016.
- [17] C. Bazgan, L. Brankovic, K. Casel, H. Fernau, K. Jansen, K. Klein, M. Lampis, M. Liedloff, J. Monnot, and V. T. Paschos. The many facets of upper domination. *Theor. Comput. Sci.*, 717:2–25, 2018.
- [18] C. Bazgan, L. Brankovic, K. Casel, H. Fernau, K. Jansen, K.-M. Klein, M. Lampis, M. Liedloff, J. Monnot, and V. T. Paschos. Upper domination: Complexity and approximation. In *International Workshop on Combinatorial Algorithms*, pages 241–252. Springer, 2016.
- [19] A. Berger, T. Fukunaga, H. Nagamochi, and O. Parekh. Approximability of the capacitated b-edge dominating set problem. *Theor. Comput. Sci.*, 385(1-3):202–213, 2007.
- [20] A. Berger and O. Parekh. Linear time algorithms for generalized edge dominating set problems. *Algorithmica*, 50(2):244–254, 2008.
- [21] P. Berman, M. Karpinski, and A. D. Scott. Approximation hardness of short symmetric instances of MAX-3SAT. *Electronic Colloquium on Computational Complexity (ECCC)*, (049), 2003.
- [22] A. A. Bertossi. Dominating sets for split and bipartite graphs. *Inf. Process. Lett.*, 19(1):37–40, 1984.
- [23] M. Biró, M. Hujter, and Z. Tuza. Precoloring extension. i. interval graphs. *Discrete Mathematics*, 100(1-3):267–279, 1992.

- [24] A. Björklund, T. Husfeldt, and N. Taslaman. Shortest cycle through specified elements. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1747–1753, 2012.
- [25] T. Bläsius, T. Friedrich, K. Meeks, and M. Schirneck. On the enumeration of minimal hitting sets in lexicographical order. *arXiv preprint arXiv:1805.01310*, 2018.
- [26] H. Böckenhauer, J. Hromkovic, R. Klasing, S. Seibert, and W. Unger. Approximation algorithms for the TSP with sharpened triangle inequality. *Inf. Process. Lett.*, 75(3):133–138, 2000.
- [27] H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on computing*, 25(6):1305–1317, 1996.
- [28] H. L. Bodlaender. Treewidth: Algorithmic techniques and results. In *International Symposium on Mathematical Foundations of Computer Science*, pages 19–36. Springer, 1997.
- [29] K. S. Booth and J. H. Johnson. Dominating sets in chordal graphs. *SIAM Journal on Computing*, 11(1):191–199, 1982.
- [30] N. Boria, F. Della Croce, and V. T. Paschos. On the max min vertex cover problem. *Discrete Applied Mathematics*, 196:62–71, 2015.
- [31] E. Boros, K. M. Elbassioni, V. Gurvich, L. Khachiyan, and K. Makino. On generating all minimal integer solutions for a monotone system of linear inequalities. In F. Orejas, P. G. Spirakis, and J. van Leeuwen, editors, *Automata, Languages and Programming, 28th International Colloquium, ICALP*, Lecture Note in Computer Science, pages 92–103.
- [32] E. Boros, V. Gurvich, and P. L. Hammer. Dual subimplicants of positive Boolean functions. *Optimization Methods and Software*, 10(2):147–156, 1998.
- [33] E. Boros, V. Gurvich, L. Khachiyan, and K. Makino. Dual-bounded generating problems: Partial and multiple transversals of a hypergraph. *SIAM Journal on Computing*, 30(6):2036–2050, 2000.
- [34] N. Bourgeois, F. D. Croce, B. Escoffier, and V. T. Paschos. Fast algorithms for min independent dominating set. *Discrete Applied Mathematics*, 161(4-5):558–572, 2013.
- [35] A. Brandstadt, J. P. Spinrad, et al. *Graph classes: a survey*, volume 3. Siam, 1999.
- [36] L. Cai. Parameterized complexity of cardinality constrained optimization problems. *The Computer Journal*, 51(1):102–121, 2008.
- [37] L. Cai, J. Chen, R. G. Downey, and M. R. Fellows. On the parameterized complexity of short computation and factorization. *Archive for Mathematical Logic*, 36(4-5):321–337, 1997.

-
- [38] J. Cardinal and E. Levy. Connected vertex covers in dense graphs. *Theor. Comput. Sci.*, 411(26-28):2581–2590, 2010.
- [39] K. Casel, J. Dreier, H. Fernau, M. Gobbert, P. Kuinke, F. S. Villaamil, M. L. Schmid, and E. J. van Leeuwen. Complexity of independency and cliquy trees. *Discrete Applied Mathematics (in press)*, 2019.
- [40] K. Casel, H. Fernau, M. K. Ghadikolaei, J. Monnot, and F. Sikora. Extension of vertex cover and independent set in some classes of graphs and generalizations. *CIAC 2019, LNCS (accepted)*, see also *CoRR*, abs/1810.04629, 2018.
- [41] K. Casel, H. Fernau, M. K. Ghadikolaei, J. Monnot, and F. Sikora. On the complexity of solution extension of optimization problems. *CoRR*, abs/1810.04553, 2018.
- [42] M. Cesati. The turing way to parameterized complexity. *Journal of Computer and System Sciences*, 67(4):654–685, 2003.
- [43] M. Cesati and M. Dilanni. Computation models for parameterized complexity. *Mathematical Logic Quarterly*, 43(2):179–202, 1997.
- [44] D. Chakrabarty and G. Goel. On the approximability of budgeted allocations and improved lower bounds for submodular welfare maximization and GAP. *SIAM Journal on Computing*, 39(6):2189–2211, 2010.
- [45] P. Chalermsook, B. Laekhanukit, and D. Nanongkai. Graph products revisited: Tight approximation hardness of induced matching, poset dimension and more. In S. Khanna, editor, *Proc. of the 24th SODA*, pages 1557–1576. SIAM, 2013.
- [46] G. J. Chang. The weighted independent domination problem is np-complete for chordal graphs. *Discrete Applied Mathematics*, 143(1-3):351–352, 2004.
- [47] M.-S. Chang. Efficient algorithms for the domination problems on interval and circular-arc graphs. *SIAM Journal on computing*, 27(6):1671–1694, 1998.
- [48] S. Chaplick, P. Dorbec, J. Kratochvíl, M. Montassier, and J. Stacho. Contact representations of planar graphs: Extending a partial representation is hard. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 139–151. Springer, 2014.
- [49] N. Chen, R. Engelberg, C. T. Nguyen, P. Raghavendra, A. Rudra, and G. Singh. Improved approximation algorithms for the spanning star forest problem. *Algorithmica*, 65(3):498–516, 2013.
- [50] M. Chlebík and J. Chlebíková. Approximation hardness of edge dominating set problems. *J. Comb. Optim.*, 11(3):279–290, 2006.
- [51] M. Chlebík and J. Chlebíková. Complexity of approximating bounded variants of optimization problems. *Theor. Comput. Sci.*, 354(3):320–338, 2006.

- [52] N. Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Management Sciences Research Group, 1976.
- [53] E. J. Cockayne and S. T. Hedetniemi. Disjoint independent dominating sets in graphs. *Discrete Mathematics*, 15(3):213–222, 1976.
- [54] C. J. Colbourn. The complexity of completing partial latin squares. *Discrete Applied Mathematics*, 8(1):25–30, 1984.
- [55] S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing (STOC 1971)*, pages 151–158. ACM, 1971.
- [56] D. G. Corneil and J. M. Keil. A dynamic programming approach to the dominating set problem on k -trees. *SIAM Journal on Algebraic Discrete Methods*, 8(4):535–543, 1987.
- [57] A. Cornet and C. Laforest. Graph problems with obligations. In *Combinatorial Optimization and Applications - 12th International Conference, COCOA 2018, Atlanta, GA, USA, December 15-17, 2018, Proceedings*, pages 183–197, 2018.
- [58] M. Cygan, F. V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized algorithms*, volume 3. Springer, 2015.
- [59] P. Damaschke. Parameterized enumeration, transversals, and imperfect phylogeny reconstruction. *Theoretical Comput. Sci.*, 351(3):337–350, 2006.
- [60] P. Damaschke, H. Müller, and D. Kratsch. Domination in convex and chordal bipartite graphs. *Inf. Process. Lett.*, 36(5):231–236, 1990.
- [61] M. Damian-Iordache and S. V. Pemmaraju. Hardness of approximating independent domination in circle graphs. In *International Symposium on Algorithms and Computation*, pages 56–69. Springer, 1999.
- [62] A. Darmann, U. Pferschy, J. Schauer, and G. J. Woeginger. Paths, trees and matchings under disjunctive constraints. *Discrete Applied Mathematics*, 159(16):1726–1735, 2011.
- [63] F. K. H. A. Dehne, M. R. Fellows, H. Fernau, E. Prieto-Rodriguez, and F. A. Rosamond. NONBLOCKER: parameterized algorithmics for minimum dominating set. In *Proc. of the 32nd SOFSEM*, volume 3831 of LNCS, pages 237–245. Springer, 2006.
- [64] F. Delbot, C. Laforest, and R. Phan. Graphs with Forbidden and Required Vertices. In *ALGOTEL 2015 - 17èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications*, Beaune, France, June 2015.
- [65] Z. Deniz, V. E. Levit, and E. Mandrescu. On graphs admitting two disjoint maximum independent sets. *CoRR*, abs/1807.06914, 2018.

-
- [66] R. G. Downey and M. R. Fellows. Fixed-parameter intractability. In *Structure in Complexity Theory Conference, 1992., Proceedings of the Seventh Annual*, pages 36–49. IEEE, 1992.
- [67] R. G. Downey and M. R. Fellows. Fixed-parameter tractability and completeness. *Congressus Numerantium*, pages 161–161, 1992.
- [68] R. G. Downey and M. R. Fellows. Fixed-parameter tractability and completeness i: Basic results. *SIAM Journal on Computing*, 24(4):873–921, 1995.
- [69] R. G. Downey and M. R. Fellows. Fixed-parameter tractability and completeness ii: On completeness for w [1]. *Theoretical Computer Science*, 141(1-2):109–131, 1995.
- [70] R. G. Downey and M. R. Fellows. Parameterized computational feasibility. In *Feasible mathematics II*, pages 219–244. Springer, 1995.
- [71] R. G. Downey and M. R. Fellows. *Parameterized complexity*. Springer, 2012.
- [72] R. G. Downey, M. R. Fellows, B. M. Kapron, M. T. Hallett, and H. T. Wareham. The parameterized complexity of some problems in logic and linguistics. In *International Symposium on Logical Foundations of Computer Science*, pages 89–100. Springer, 1994.
- [73] S. Dudycz, M. Lewandowski, and J. Marcinkowski. Tight approximation ratio for minimum maximal matching. *arXiv preprint arXiv:1811.08506*, 2018.
- [74] B. Escoffier, L. Gourvès, and J. Monnot. Complexity and approximation results for the connected vertex cover problem in graphs and hypergraphs. *Journal of Discrete Algorithms*, 8(1):36–49, 2010.
- [75] B. Escoffier, J. Monnot, V. T. Paschos, and M. Xiao. New results on polynomial inapproximability and fixed parameter approximability of edge dominating set. *Theory of Computing Systems*, 56(2):330–346, 2015.
- [76] M. Farber. Independent domination in chordal graphs. *Operations Research Letters*, 1(4):134–138, 1982.
- [77] M. Farber. Domination, independent domination, and duality in strongly chordal graphs. *Discrete Applied Mathematics*, 7(2):115–130, 1984.
- [78] M. R. Fellows. *Parameterized complexity*. Springer-Verlag New York Incorporated, 1999.
- [79] M. R. Fellows, D. Hermelin, F. Rosamond, and S. Vialette. On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science*, 410(1):53–61, 2009.
- [80] H. Fernau. On parameterized enumeration. In *International Computing and Combinatorics Conference*, pages 564–573. Springer, 2002.

- [81] H. Fernau. Parameterized algorithmics: A graph-theoretic approach. *Germany: Habilitationsschrift, Universität Tübingen*, 2005.
- [82] H. Fernau. Edge dominating set: Efficient enumeration-based exact algorithms. In *International Workshop on Parameterized and Exact Computation*, pages 142–153. Springer, 2006.
- [83] H. Fernau and D. F. Manlove. Vertex and edge covers with clustering properties: Complexity and algorithms. *Journal of Discrete Algorithms*, 7(2):149–167, 2009.
- [84] D. Fotakis, L. Gourvès, and J. Monnot. Conference program design with single-peaked and single-crossing preferences. In *Web and Internet Economics - 12th International Conference, WINE 2016, Montreal, Canada, December 11-14, 2016, Proceedings*, pages 221–235, 2016.
- [85] H. N. Gabow. An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, pages 448–456, 1983.
- [86] L. Gai and G. Zhang. On lazy bureaucrat scheduling with common deadlines. *J. Comb. Optim.*, 15(2):191–199, 2008.
- [87] M. R. Garey and D. S. Johnson. The rectilinear steiner tree problem is np-complete. *SIAM Journal on Applied Mathematics*, 32(4):826–834, 1977.
- [88] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979.
- [89] F. Gavril. Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph. *SIAM Journal on Computing*, 1(2):180–187, 1972.
- [90] F. Gavril. Some np-complete problems on graphs. In *Proc. Conf. on Inform. Sci. and Systems, 1977*, pages 91–95, 1977.
- [91] M. Gendreau, G. Laporte, and A. Hertz. An approximation algorithm for the traveling salesman problem with backhauls. *Operations Research*, 45(4):639–641, 1997.
- [92] P. A. Golovach, P. Heggernes, M. M. Kanté, D. Kratsch, and Y. Villanger. Enumerating minimal dominating sets in chordal bipartite graphs. *Discrete Applied Mathematics*, 199:30–36, 2016.
- [93] P. A. Golovach, P. Heggernes, M. M. Kanté, D. Kratsch, and Y. Villanger. Minimal dominating sets in interval graphs and trees. *Discrete Applied Mathematics*, 216:162–170, 2017.

- [94] P. A. Golovach, P. Heggernes, and D. Kratsch. Enumeration and maximum number of minimal connected vertex covers in graphs. *European Journal of Combinatorics*, 68:132–147, 2018. Combinatorial Algorithms, Dedicated to the Memory of Mirka Miller.
- [95] P. A. Golovach, P. Heggernes, D. Kratsch, and Y. Villanger. An incremental polynomial time algorithm to enumerate all minimal edge dominating sets. *Algorithmica*, 72(3):836–859, 2015.
- [96] L. Gourvès, J. Monnot, and A. Pagourtzis. The lazy matroid problem. In *Prof. of the 8th IFIP TCS*, volume 8705 of *LNCS*, pages 66–77. Springer, 2014.
- [97] L. Gourvès, J. Monnot, and L. Tlilane. Subset sum problems with digraph constraints. *J. Comb. Optim.*, 36(3):937–964, 2018.
- [98] Q.-P. Gu and H. Tamaki. Improved bounds on the planar branchwidth with respect to the largest grid minor size. *Algorithmica*, 64(3):416–453, 2012.
- [99] N. Guttmann-Beck, R. Hassin, S. Khuller, and B. Raghavachari. Approximation algorithms with bounded performance guarantees for the clustered traveling salesman problem. *Algorithmica*, 28(4):422–437, 2000.
- [100] M. M. Halldórsson. Approximating the minimum maximal independence number. *Information Processing Letters*, 46(4):169–172, 1993.
- [101] J. Håstad. Some optimal inapproximability results. *Journal of the ACM (JACM)*, 48(4):798–859, 2001.
- [102] R. B. Hayward. Weakly triangulated graphs. *J. Comb. Theory, Ser. B*, 39(3):200–208, 1985.
- [103] J. He and H. Liang. On variants of the spanning star forest problem. In *Frontiers in Algorithmics and Algorithmic Aspects in Information and Management - Joint International Conference, FAW-AAIM 2011, Jinhua, China, May 28-31, 2011. Proceedings*, pages 70–81, 2011.
- [104] J. He and H. Liang. Improved approximation for spanning star forest in dense graphs. *J. Comb. Optim.*, 25(2):255–264, 2013.
- [105] M. A. Henning, C. Löwenstein, and D. Rautenbach. Remarks about disjoint dominating sets. *Discrete Mathematics*, 309(23-24):6451–6458, 2009.
- [106] S. D. Hochbaum. *Approximation Algorithms for NP-hard Problems*. PWS Publishing Co., 1997.
- [107] A. Hujdurovic, M. Milanic, and B. Ries. Detecting strong cliques. *CoRR*, abs/1808.08817, 2018.

- [108] A. Hujdurovic, M. Milanic, and B. Ries. Graphs vertex-partitionable into strong cliques. *Discrete Mathematics*, 341(5):1392–1405, 2018.
- [109] M. Hujter and Z. Tuza. Precoloring extension. ii. graph classes related to bipartite graphs. *Acta Mathematica Universitatis Comenianae*, 62(1):1–11, 1993.
- [110] R. Impagliazzo and R. Paturi. Complexity of k-sat. In *Fourteenth Annual IEEE Conference on Computational Complexity*, pages 237–240. IEEE, 1999.
- [111] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.
- [112] R. W. Irving and D. Manlove. The b-chromatic number of a graph. *Discrete Applied Mathematics*, 91(1-3):127–141, 1999.
- [113] K. Jansen. An approximation algorithm for the general routing problem. *Inf. Process. Lett.*, 41(6):333–339, 1992.
- [114] V. Kann. *On the approximability of NP-complete optimization problems*. PhD thesis, Royal Institute of Technology Stockholm, 1992.
- [115] M. M. Kanté, V. Limouzy, A. Mary, and L. Nourine. On the enumeration of minimal dominating sets and related notions. *SIAM Journal of Discrete Mathematics*, 28(4):1916–1929, 2014.
- [116] M. M. Kanté, V. Limouzy, A. Mary, L. Nourine, and T. Uno. Polynomial delay algorithm for listing minimal edge dominating sets in graphs. In F. Dehne, J. Sack, and U. Stege, editors, *Workshop on Algorithms and Data Structures, WADS*, volume 9214 of *Lecture Note in Computer Science*, pages 446–457. Springer, 2015.
- [117] M. M. Kanté, V. Limouzy, A. Mary, L. Nourine, and T. Uno. A polynomial delay algorithm for enumerating minimal dominating sets in chordal graphs. In E. W. Mayr, editor, *International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2015*, volume 9224, pages 138–153. Springer, 2016.
- [118] R. M. Karp. Reducibility among combinatorial problems. In *Proceedings of a symposium on the Complexity of Computer Computations, IBM Thomas J. Watson Research Center, Yorktown Heights*, pages 85–103, 1972.
- [119] L. Khachiyan, E. Boros, K. M. Elbassioni, and V. Gurvich. On enumerating minimal dicuts and strongly connected subgraphs. *Algorithmica*, 50(1):159–172, 2008.
- [120] L. G. Khachiyan, E. Boros, K. M. Elbassioni, V. Gurvich, and K. Makino. On the complexity of some enumeration problems for matroids. *SIAM Journal of Discrete Mathematics*, 19(4):966–984, 2005.

-
- [121] P. Klavík, J. Kratochvíl, T. Krawczyk, and B. Walczak. Extending partial representations of function graphs and permutation graphs. In *European Symposium on Algorithms*, pages 671–682. Springer, 2012.
- [122] P. Klavík, J. Kratochvíl, Y. Otachi, I. Rutter, T. Saitoh, M. Saumell, and T. Vyskočil. Extending partial representations of proper and unit interval graphs. *Algorithmica*, 77(4):1071–1104, 2017.
- [123] P. Klavík, J. Kratochvíl, Y. Otachi, and T. Saitoh. Extending partial representations of subclasses of chordal graphs. *Theoretical Computer Science*, 576:85–101, 2015.
- [124] P. Klavík, J. Kratochvíl, Y. Otachi, T. Saitoh, and T. Vyskočil. Extending partial representations of interval graphs. *Algorithmica*, 78(3):945–967, 2017.
- [125] T. Kloks. *Treewidth: computations and approximations*, volume 842. Springer Science & Business Media, 1994.
- [126] L. Kowalik and M. Mucha. A 9k kernel for nonseparating independent set in planar graphs. *Theor. Comput. Sci.*, 516:86–95, 2014.
- [127] J. Kratochvíl. A special planar satisfiability problem and a consequence of its np-completeness. *Discrete Applied Mathematics*, 52(3):233–252, 1994.
- [128] J. Kratochvíl, A. Proskurowski, and J. A. Telle. Covering regular graphs. *Journal of Combinatorial Theory, Series B*, 71(1):1–16, 1997.
- [129] J. Kratochvíl, A. Proskurowski, and J. A. Telle. On the complexity of graph covering problems. *Nord. J. Comput.*, 5(3):173–195, 1998.
- [130] R. Krithika, D. Majumdar, and V. Raman. Revisiting connected vertex cover: FPT algorithms and lossy kernels. *Theory Comput. Syst.*, 62(8):1690–1714, 2018.
- [131] E. L. Lawler, J. K. Lenstra, and A. H. G. R. Kan. Generating all maximal independent sets: NP-hardness and polynomial-time algorithms. *SIAM Journal on Computing*, 9:558–565, 1980.
- [132] D. Lokshtanov, D. Marx, and S. Saurabh. Known algorithms on graphs of bounded treewidth are probably optimal. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 777–789. Society for Industrial and Applied Mathematics, 2011.
- [133] D. Lokshtanov, D. Marx, and S. Saurabh. Slightly superexponential parameterized problems. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 760–776. Society for Industrial and Applied Mathematics, 2011.
- [134] L. Lovász. Coverings and colorings of hypergraphs. In *Proc. 4th Southeastern Conference of Combinatorics, Graph Theory, and Computing*, pages 3–12. Utilitas Mathematica Publishing, 1973.

- [135] D. Manlove. On the 2-maximal independence number of a graph. Technical report, University of Glasgow, Department of Computing Science, 1997.
- [136] D. F. Manlove. *Minimaximal and maximinimal optimisation problems: a partial order-based approach*. PhD thesis, University of Glasgow, 1998.
- [137] D. F. Manlove. On the algorithmic complexity of twelve covering and independence parameters of graphs. *Discrete Applied Mathematics*, 91(1-3):155–175, 1999.
- [138] D. Marx. Np-completeness of list coloring and precoloring extension on the edges of planar graphs. *Journal of Graph Theory*, 49(4):313–324, 2005.
- [139] D. Marx. Precoloring extension on unit interval graphs. *Discrete Applied Mathematics*, 154(6):995–1002, 2006.
- [140] D. Marx. Parameterized complexity and approximation algorithms. *The Computer Journal*, 51(1):60–78, 2008.
- [141] S. Mishra and K. Sikdar. On the hardness of approximating some np-optimization problems related to minimum linear ordering problem. In *IFIP International Conference on Theoretical Computer Science*, pages 186–199. Springer, 2000.
- [142] J. W. Moon and L. Moser. On cliques in graphs. *Israel Journal of Mathematics*, 3:23–28, 1965.
- [143] C. T. Nguyen, J. Shen, M. Hou, L. Sheng, W. Miller, and L. Zhang. Approximating the spanning star forest problem and its application to genomic sequence alignment. *SIAM Journal on Computing*, 38(3):946–962, 2008.
- [144] V. H. Nguyen. The maximum weight spanning star forest problem on cactus graphs. *Discrete Math., Alg. and Appl.*, 7(2), 2015.
- [145] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.
- [146] C. Ojima, A. Saito, and K. Sano. Precoloring extension involving pairs of vertices of small distance. *Discrete Applied Mathematics*, 166:170–177, 2014.
- [147] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of computer and system sciences*, 43(3):425–440, 1991.
- [148] P. Priyadarsini and T. Hemalatha. Connected vertex cover in 2-connected planar graph with maximum degree 4 is NP-complete. *International Journal of Mathematical, Physical and Engineering Sciences*, 2(1):51–54, 2008.
- [149] C. Savage. Depth-first search and the vertex cover problem. *Information Processing Letters*, 14(5):233–235, 1982.

-
- [150] O. Schaudt. On disjoint maximal independent sets in graphs. *Inf. Process. Lett.*, 115(1):23–27, 2015.
- [151] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer-Verlag Berlin and Heidelberg GmbH & Co., 2003.
- [152] B. Schwikowski and E. Speckenmeyer. On enumerating all minimal solutions of feedback problems. *Discrete Applied Mathematics*, 117:253–265, 2002.
- [153] E. Speckenmeyer. On feedback vertex sets and nonseparating independent sets in cubic graphs. *Journal of Graph Theory*, 12(3):405–412, 1988.
- [154] L. Trevisan. Non-approximability results for optimization problems on bounded degree instances. In *Proc. of the 33rd STOC*, pages 453–461. ACM, 2001.
- [155] R. Uehara. Tractable and intractable problems on generalized chordal graphs. Technical report, Technical Report COMP98-83, IEICE, 1999.
- [156] S. Ueno, Y. Kajitani, and S. Gotoh. On the nonseparating independent set problem and feedback set problem for graphs with no vertex degree exceeding three. *Discrete Mathematics*, 72(1-3):355–360, 1988.
- [157] T. Uno. Algorithms for enumerating all perfect, maximum and maximal matchings in bipartite graphs. In *Algorithms and Computation, 8th International Symposium, ISAAC '97, Singapore, December 17-19, 1997, Proceedings*, pages 92–101, 1997.
- [158] V. V. Vazirani. *Approximation Algorithms*. Springer, 2001.
- [159] M. Voigt. Precoloring extension for 2-connected graphs. *SIAM Journal of Discrete Mathematics*, 21(1):258–263, 2007.
- [160] J. Wang, B. Chen, Q. Feng, and J. Chen. An efficient fixed-parameter enumeration algorithm for weighted edge dominating set. In *Frontiers in Algorithmics, Third International Workshop, FAW 2009, Hefei, China, June 20-23, 2009. Proceedings*, pages 237–250, 2009.
- [161] M. Weller, A. Chateau, R. Giroudeau, J. König, and V. Pollet. On residual approximation in solution extension problems. *J. Comb. Optim.*, 36(4):1195–1220, 2018.
- [162] M. Yannakakis and F. Gavril. Edge dominating sets in graphs. *SIAM Journal on Applied Mathematics*, 38(3):364–372, 1980.
- [163] M. Zehavi. Maximum minimal vertex cover parameterized by vertex cover. *SIAM Journal on Discrete Mathematics*, 31(4):2440–2456, 2017.
- [164] D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 681–690. ACM, 2006.

BIBLIOGRAPHY

- [165] D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3(1):103–128, 2007.

RÉSUMÉ

Le problème de la détermination de la qualité d'une solution partielle se pose dans la majeure partie des approches algorithmiques cherchant à calculer progressivement une solution globale. L'élagage des arbres de recherche, la preuve de garanties d'approximation et l'efficacité des stratégies d'énumération sont des approches algorithmiques qui exigent souvent un moyen approprié de décider si une solution partielle donnée est un bon candidat pour l'étendre à une solution globale de bonne qualité.

Dans cette thèse, nous étudions un type particulier de problèmes d'optimisation, appelés *problèmes d'extension* pour un grand nombre de problèmes basés sur des graphes. Contredisant peut-être l'intuition, ces problèmes ont tendance à être NP-difficile, même quand le problème d'optimisation sous-jacent peut être résolu en temps polynomial. Nous présentons de nombreux résultats positifs et négatifs de NP-difficulté et d'approximation pour différents scénarios d'entrée. De plus, nous étudions la complexité paramétrée des problèmes d'extension par rapport à la taille des pré-solutions, ainsi que l'optimalité de certains algorithmes exacts sous l'hypothèse du temps exponentielle.

MOTS CLÉS

Complexité, Approximation, Complexité paramétrée, Problèmes d'extension, Graphe

ABSTRACT

The problem of determining the quality of a partial solution occurs in almost every algorithmic approach that gradually computes a global solution. Pruning search trees, proving approximation guarantees, or the efficiency of enumeration strategies usually require a suitable way to decide if a given partial solution is a reasonable candidate to pursue for extension to a global one, of assured quality.

In this thesis, we study a special type of optimization problems, called *extension problems* for a large number of optimization problems on graphs. Possibly contradicting intuition, these problems tend to be NP-hard, even for problems where the underlying optimization problem can be solved in polynomial time. We present many positive/negative hardness and approximation results for different input scenarios. Moreover, the parameterized complexity of extension problems with respect to size of partial solutions, as well as the optimality of some exact algorithms under the Exponential-Time Hypothesis (ETH) are studied.

KEYWORDS

Computational complexity, Approximation, Parameterized complexity, Extension problems, Graph problems