A STUDY OF THE BICLIQUE EDGE PARTITION AND COVER PROBLEMS

Hideaki OTSUKI

Acknowledgements

I would like to express my sincere gratitude to my advisor Prof. Tomio Hirata for the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Harumichi Nishimura, Prof. Masahiko Sakai, and Prof. Mutsunori Yagiura for their encouragement, insightful comments, and hard questions.

Contents

Abstract

This thesis addresses the biclique edge partition (cover) problem, the problem of finding the minimum number of bicliques to partition (cover) the edge set of a given input bipartite graph.

The biclique edge partition problem is known to be NP-hard, and few results are known about its approximation hardness. In this thesis, a lower bound for the approximation ratio is demonstrated. Specifically, it is shown that we cannot have an approximate solution that is less than 6053/6052 times the optimal solution unless P = NP. To obtain this hardness result, a gap-preserving reduction from the satisfiability problem to the biclique edge partition problem is given.

The biclique edge cover problem is also known to be NP-hard, and a polynomial time algorithm has been proposed for some restricted graph classes. For C_4-free graphs and distance-hereditary bipartite graphs, a polynomial time algorithm was given in 1996. For domino-free graphs, which properly contain C_4-free and distance-hereditary bipartite graphs, a polynomial time algorithm was given in 1999. In this thesis, a polynomial time algorithm for a new graph class is proposed. This graph class properly contains the class of domino-free bipartite graphs. Specifically, for a given bipartite graph B, a parameter $R(B)$ is defined using the modified Galois lattice, $G_m(B)$. This thesis shows that B is a domino-free graph if and only if $R(B) = 0$. The graph class defined by $R(B) \leq 1$ properly includes the domino-free graphs. A polynomial time algorithm is proposed for this new graph class.

It has been shown that if B is a domino-free graph then the size of the modified Galois lattice $G_m(B)$ is $O(nm)$, where n and m are the numbers of vertices and edges of B, respectively. In this thesis, it is shown that the size of $G_m(B)$ is actually $O(n+m)$ for graphs with $R(B) = O(1)$. This improves the time complexity of the algorithm proposed in this thesis.

Chapter 1

Introduction

Graphs are not only mathematical objects but also very useful tools for modeling real-world problems. In particular, bipartite graphs are often used to model real-world problems. For example, in data mining, relations between two disjoint sets - such as "customers" and "products" or "phrases" and "documents" - are modeled by bipartite graphs. Also, in formal concept analysis, "objects" and "attributes" are well-modeled by bipartite graphs. In these applications, bipartite graphs are often required to be covered with bicliques. In this thesis, problems of finding the minimum number of bicliques to partition or cover the edges of a given bipartite simple graph are studied.

The biclique edge partition problem is the problem of finding the minimum number of bicliques (complete bipartite graphs) to partition all of the edges of a given bipartite graph, whereas the biclique edge cover problem is the problem of finding the minimum number of bicliques to cover all edges of a given bipartite graph. These problems are very similar. The difference is that the biclique edge partition problem does not allow the bicliques in its solution to share edges.

In graph theory, problems of partitioning the set of vertices are thoroughly studied. For example, the graph coloring problem is well known. This problem is equivalent to the clique partition problem, which is the problem of finding the minimum number of cliques that partition all of the vertices of a given graph, since the solution of the graph coloring problem for G is also the solution of the clique partition problem for $\bar{G}$, which is the complement of G. Note that the clique partition and the clique cover problems are essentially the same. However, when it comes to the problem of partitioning or covering the set of edges of a graph, the edge partition and edge cover problems are not the same.

The edge-partition and cover problems do not appear to have been widely studied in comparison to vertex-partition and cover problems. However, these problems have applications in many areas such as electronic circuits, networks, data-minings, computer graphics, and marketing. Let us consider an example wherein these problems are applied to marketing data analysis (Market Basket Analysis).

In supermarkets, it is often the case that some items are consistently purchased together, for example, "breads" and "butter", "coffee", "milk", and "sugar" or "wine" and "cheese" and so on. Let us say that these items are associated with each other. To gain as much profit as possible, it is best to put the associated set of items on a single shelf. But all sets of items cannot be put on a single shelf, because the size of a shelf is limited. Assume that the manager of a supermarket requires that each customer buys all items from a single shelf. The minimum number of shelves that satisfies this requirement is found by the biclique edge partition and cover problems. Suppose that a 0-1 matrix M represents the shop's transaction data such that if ith customer buys the jth item, $M_{i,j} = 1$, otherwise $M_{ij} = 0$. Let

$$M = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

be an example of such a matrix. B_M, an instance of the biclique edge partition and cover problems, is constructed from M as follows. Make a bipartite graph B_M such that there is an edge (x_i, y_j) if and only if the ith customer buys the jth item. An example of B_M is shown in Fig.1.1. In this example, customer $\{1\}$ buys items $\{1, 2\}$, customer $\{2\}$ buys items $\{1, 2, 3, 4\}$, customers $\{3, 4\}$ buy items $\{2, 3, 4\}$ and, customers $\{5, 6\}$ buy items $\{4, 5, 6\}$. It is easy to verify that the edge set of B can be partitioned into four bicliques:

$$\begin{aligned} B_1 &= (\{x_1\}, \{y_1, y_2\}), \\ B_2 &= (\{x_2\}, \{y_1, y_2, y_3, y_4\}), \\ B_3 &= (\{x_3, x_4\}, \{y_2, y_3, y_4\}), \\ B_4 &= (\{x_5, x_6\}, \{y_4, y_5, y_6\}). \end{aligned}$$

This partitioning gives the minimum number of the biclique edge partitions of B. In this case, four shelves are sufficient to satisfy the requirement. That is, each set of

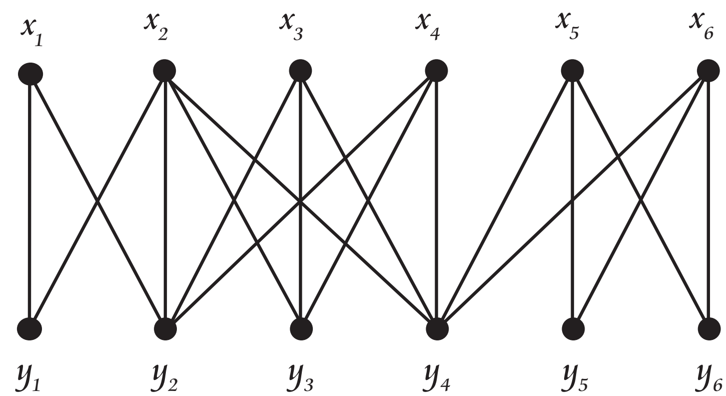

Figure 1.1: The bipartite graph B_M represents the purchasing data.

items $\{1,2\}$, $\{1,2,3,4\}$, $\{2,3,4\}$ and $\{4,5,6\}$ is put on each shelf, for a total of four shelves. Note that each customer buys no extra item. Meanwhile, B can be covered by these three bicliques,

$$\begin{aligned} B'_1 &= (\{x_1,x_2\},\{y_1,y_2\}), \\ B'_2 &= (\{x_2,x_3,x_4\},\{y_2,y_3,y_4\}), \\ B'_3 &= (\{x_5,x_6\},\{y_4,y_5,y_6\}). \end{aligned}$$

This covering information suggests that the manager of the supermarket should reduce the number of shelves by one. When there are three shelves such that each set of items $\{1,2\}$, $\{2,3,4\}$, and $\{4,5,6\}$ is put on each shelf, customer 2 has to buy two of item 2 under the manager's requirement. If customers are wiling to buy excess items, then the biclique edge cover problem gives the minimum number of shelves. This is also an example that shows that the size of minimum biclique edge partitioning is not always equal to the size of the minimum biclique edge covering.

As the biclique edge partition and cover problems are NP-hard, it is not possible to find the optimal solution in polynomial time unless P = NP. In order to solve NP-hard problems, heuristic algorithms are often used, and they provide solutions to NP-hard problems that are good enough for practical purposes. However, there is no guarantee that they will always provide good solutions in polynomial time. In contrast to heuristic algorithms, approximation algorithms provide a solution close enough to the optimal solution even in the worst case. Research on approximation algorithms has shown that not all NP-hard problems have the same approximation hardness, and that NP-hard problems can be classified by their approximation hardness under the assumption that P $\neq$ NP. The hardness of approximation is measured by an *approximation ratio* ρ. A ρ-approximation algorithm for a minimization (maximization) problem is the algorithm that is guaranteed to return a solution with a size that is at most ρ (at least $1/\rho$) times the size of the optimal solution. We say that an optimization problem has a polynomial time approximation scheme (PTAS) if it admits a ρ-approximation algorithm for any $\rho > 1$. For example, the knapsack problem has a PTAS. Problems that have a PTAS are easy to approximate, but there are a lot of NP-hard problems that do not have a PTAS. For example, the MAX 3-SAT problem has an 8/7-approximation algorithm [KZ97], but it cannot be approximated to within a ratio $8/7 - \epsilon$ for any $\epsilon > 0$ [Hås01]. Furthermore, MAX CLIQUE has no ρ-approximation algorithm with a constant ρ. Let APX denote the class of approximation problems that have ρ-approximation algorithms. Then MAX 3-SAT belongs to APX, whereas MAX CLIQUE does not.

Further research on the classification of NP-hard problems has been progressed since the discovery of the PCP theorem, which offers a new characterization of NP by the probabilistically checkable proof (PCP) system. The PCP theorem states that for NP-complete problems, there exist witnesses which can be verified with high probability by only looking at a constant number of randomly chosen bits. The PCP theorem was originally proved as a result of series of studies using the theory of codes and the interactive proof [FGL$^+$96] [AS98], and thus it was surprise to know that the PCP theorem is closely related to the proof of approximation hardness of **NP**-hard problems. In past decades, approximation hardness results for many important NP-hard problems have been proved directly or indirectly using the PCP theorem. The discovery of the approximation threshold of 8/7 for MAX 3-SAT [Hås01] and the discovery of an approximation hardness of $O(n^{1-\epsilon})$ for MAX CLIQUE [Hås99] are remarkable results obtained from the PCP theorem. With respect to the minimum biclique edge cover problem, Gruber et al. showed that there is no $O(n^{1/3-\epsilon})$ and $O(m^{1/5-\epsilon})$-approximation algorithm for any $\epsilon > 0$ [GH07]. The original proof of the PCP theorem was complicated. However, Dinur [Din07] presented a relatively simple proof of the PCP theorem using the gap-amplification method of the constraint graphs.

In this thesis, it is shown that the biclique edge partition problem cannot be approximated within a ratio of 6053/6052 unless P = NP. In order to obtain this approximation hardness, an approximation-preserving (gap-preserving) reduction from 3-OCC-MAX 2-SAT is presented. 3-OCC-MAX 2-SAT is the satisfiability problem such that each clause has exactly two literals, and every variable occurs exactly three times. Berman and Karpinski [BK98] showed that a 2012/2011-approximation algorithm for 3-OCC-MAX 2-SAT is NP-hard. Our reduction implies that if there is a 6053/6052-approximation algorithm for the biclique edge partition problem then P = NP. Therefore, under the assumption that P $\neq$ NP, the biclique edge partition problem does not have a 6053/6052-approximation algorithm. It follows that the biclique edge partition problem has no polynomial-time approximation scheme unless P = NP. To the author's knowledge, the ratio of 6053/6052 is the first explicit lower bound on the approximation hardness of the biclique edge partition problem.

Now, let us consider the computational complexity of the biclique edge cover problem. For general bipartite graphs, the biclique edge cover problem cannot be solved in polynomial time if P $\neq$ NP. However, it can be solved in polynomial time if we restrict the class of input graphs. Müller showed that there is a polynomial time algorithm for C_4-free bipartite graphs and for distance-hereditary bipartite graphs [Mül96]. A C_4-free bipartite graph is a bipartite graph that has no C_4, a cycle of length four, as an induced subgraph. The distance-hereditary bipartite graph can be characterized in several ways. One of these characterizations is that it is a $(6, 2)$-chordal bipartite graph. A bipartite graph is $(6, 2)$-chordal if every cycle of length at least six has at least two chords. A bipartite graph is domino-free if it has no domino as an induced subgraph. Amilhastre et al. showed that there is a polynomial time algorithm for domino-free bipartite graphs [AVJ98]. The class of domino-free graphs properly contains both the C_4-free and the distance-hereditary bipartite graph classes. It is obvious that a solution to the biclique edge partition problem is also a solution to the biclique edge cover problem. Amilhastre et al. showed that a solution to the biclique edge partition problem can be obtained from a solution to the biclique edge cover problem without changing the solution size if the graph is domino-free. Thus the minimum biclique edge partition problem can also be solved in polynomial time for a domino-free bipartite graph. This thesis extends the graph class for which the minimum biclique edge cover problem can be solved in polynomial time.

This thesis defines a new graph class for which the minimum biclique edge cover problem can be solved in polynomial time, and shows that this graph class properly contains the domino-free graph class. In order to present the new graph class, the modified Galois lattice $G_m(B)$ for an input bipartite graph B is introduced. A partial order on the set of maximal bicliques in B is defined, and $G_m(B)$ is a Hasse diagram of this partial order. Furthermore, the redundant parameter $R(B)$ is defined on $G_m(B)$. It is shown that $R(B) = 0$ if and only if B is domino free. Furthermore, it is shown that there is a polynomial time algorithm for a graph B with $R(B) = 1$. If $R(B) > 0$, then B has at least one domino as an induced subgraph. Thus, we have a new graph class such that there is a polynomial time algorithm for the minimum biclique edge cover problem. See Fig. 1.2.

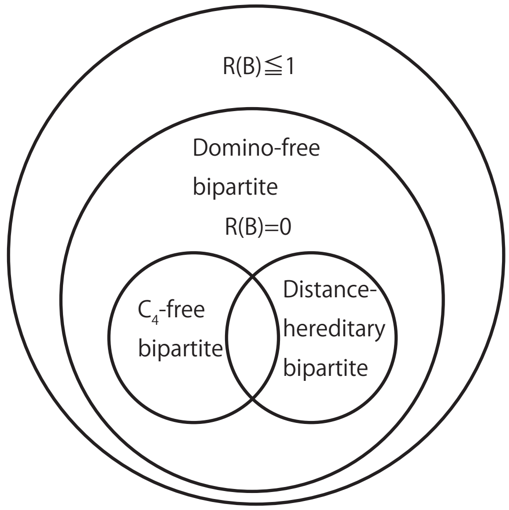

Figure 1.2: The new graph class with $R(B) = 1$ properly includes the domino-free graph class.

The computation time of the proposed algorithm depends on the size of $G_m(B)$. The size of $G_m(B)$ could be as large as exponential in the size of B for a general bipartite graph, B. However, this thesis shows that $G_m(B)$ has at most $2n + 1$ vertices for a distance-hereditary bipartite graph, B. For a graph B such that $R(B) \leq 1$, it is shown that $G_m(B)$ has at most $O(n + m)$ vertices.

The structure of this thesis is as follows. In Chapter 2, definitions, notations, and backgrounds of this study are given. In Chapter 3, the approximation hardness result for the biclique edge partition problem is shown. In Chapter 4, the modified Galois lattice $G_m(B)$ is defined, and the redundant parameter $R(B)$ is introduced. It is proved that for a graph with $R(B) \leq 1$, the biclique edge cover problem can be solved in polynomial time. In Chapter 5, the size of the modified Galois lattice for some restricted graph classes is investigated. Chapter 6 is the conclusion.

Chapter 2

Preliminary

As a preliminary for this thesis, basic terms and definitions are presented in this chapter. In order to introduce the concept of computational complexity class, basic terms (alphabet, language, Turing machines, and so on) are defined in Sections 2.1, 2.2, and 2.3. Basic terms in the graph theory are introduced in Section 2.4. Satisfiability problems has important role in the theory of computational complexity and approximation algorithms. These backgrounds are noticed from Sections 2.5 to 2.10. In Section 2.11, polynomial time algorithms for the biclique edge cover and partition problems for restricted graph classes are reviewed.

2.1 Alphabet, relation, and partial order

Definition 1 (Alphabet, String)**.** *The alphabet* Σ *is a finite set of symbols or letters. A sequence of symbols in* Σ

$$a_1 a_2 \cdots a_n \ (a_1, a_2, \ldots, a_n \in \Sigma)$$

is the string or word on Σ*. The concatenation of strings* $a = a_1 a_2 \cdots a_n$ *and* $b = b_1 b_2 \cdots b_m$ *is* $ab = a_1 a_2 \cdots a_n b_1 b_2 \cdots b_m$*.*

The *length* of $w = a_1 a_2 \cdots a_n$ is the number of symbols in w, denoted by $|w| (= n)$. If $|w| = 0$ then w is the *empty word* denoted by ϵ.

Definition 2 (Language)**.** Σ^* *is the set of all strings on* Σ*.* Σ^+ *is* $\Sigma^* \setminus \{\epsilon\}$*. An language on* Σ *is a subset of* Σ^**.*

For languages L and L' on Σ, $LL' = \{xy \mid x \in L, y \in L'\}$ is the concatenation of L and L'.

Definition 3 (Kleene closure)**.** *L^n is the concatenation of n Ls. Kleene closure L^* is $\bigcup_{n\geq 0} L^n$. Note that L^0 is $\{\epsilon\}$. Kleene plus L^+ is $\bigcup_{n\geq 1} L^n$.*

Definition 4 (Binary relation)**.** *A binary relation R of two sets X and Y is a subset of $X \times Y$. $R : X \to Y$ means that R is a subset of $X \times Y$.*

Definition 5 (k-ary relation)**.** *A k-ary relation of k sets X_i $(i = 1, \ldots, k)$ is a subset of $X_1 \times \cdots \times X_k$.*

Throughout this thesis, a relation is a binary relation unless otherwise noticed.

R is a relation on X if $R : X \to X$. If $(x, y) \in R$ then x and y *have a relation* R denoted by xRy. $R : X \to Y$ is a *partial function* from X to Y if for all $x \in X$ there is at most one $y \in Y$. R is a *total function* (or simply a *function*) or a *mapping* from X to Y if for all $x \in X$ there is exactly one $y \in Y$. For a partial function R, $(x, y) \in R$ is denoted by $R(x) = y$. Let $R : X \to Y$ and $S : Y \to Z$. The *composition* of R and S is a relation $R \circ S : X \to Z$ defined by

$$R \circ S = \{(x, z) \mid \exists y \in Y, (x, y) \in R, (y, z) \in S\}.$$

For an integer $n \geq 0$ and a relation R on X, relations R^n are defined as follows:

$$R^0 = \{(x, x) \mid x \in X\},\ \ R^{n+1} = R^n \circ R.$$

$R^* = \bigcup_{n\geq 0} R^n$ is a *reflexive transitive closure*, $R^+ = \bigcup_{n\geq 1} R^n$ is a *transitive closure* of R.

Definition 6 (Partial order)**.** *The partial order on X is a relation R if for any $x, y, z \in X$, R satisfies following three properties: (i) xRx (reflexive law), (ii) xRy and $yRz \Rightarrow xRz$ (transitive law), (iii) xRy and $yRx \Rightarrow x = y$ (antisymmetric law). A partial order is often denoted by $\leq$.*

A *partially ordered set* is a set on which a partial order is defined. $(X, \leq)$ means that X is a partially ordered set of $\leq$.

2.2 Turing machines

Definition 7 (Turing machine)**.** *A Turing machine is a quadruple $M = (K, \Sigma, \delta, s)$. K is a finite set of states and s is the initial state. Σ is the alphabet of M. Here K and Σ are disjoint sets. Σ contains two of the special symbols the blank $\sqcup$ and the first* *symbol* $\triangleright$. *δ is a transition function, which maps $K \times \Sigma$ to $(K \cup \{h, \text{"yes"}, \text{"no"}\}) \times \Sigma \times \{\leftarrow, \rightarrow, -\}$. h is the halting state, "yes" and "no" is the accepting state and rejecting state respectively. $\leftarrow$ and $\rightarrow$ is the cursor direction for "left" and "right" respectively, and $-$ for stay. Note that $\leftarrow$, $\rightarrow$ and $-$ are not in $K \cup \Sigma$. Let current state $q \in K$ and current symbol $\sigma \in \Sigma$. Function δ specifies for each combination of q and σ, a triple $\delta(q, \sigma) = (p, \rho, D)$. Here p is the next state, ρ is the symbol to be overwritten on σ, and $D \in \{\leftarrow, \rightarrow, -\}$ is the direction in which the cursor will move. For any state q, $\delta(q, \triangleright) = (\triangleright, p, \rightarrow)$. That is, $\triangleright$ always directs the cursor to the right, and is never erased.*

With the initial state s, a Turing machine works as follows. The string is initialized to a $\triangleright$, followed by a finitely long string $x \in (\Sigma \setminus \{\sqcup\})^*$. Here x is the *input* of the Turing machine. The cursor is pointing the first symbol, always a $\triangleright$. From this initial configuration the machine takes a step according to δ, changing its state, overwriting a symbol, and moving the cursor. Note that the string will always start with a $\triangleright$, and thus the cursor will never "fall off" the left end of the string. The machine has "*halted*" if one of the three states h, "yes" or "no" has reached. Furthermore, the machine *accepts* the input if state "yes" has been reached, and it *rejects* the input if state "no" has been reached. If a machine halts on input x, the *output* of the machine M on x can be defined. In this case, the output is denoted by $M(x)$. If M accepts or rejects x then $M(x) = \text{"yes"}$ or $M(x) = \text{"no"}$ respectively. Otherwise, if h was reached, then the output is the string of M at the time of halting. Since the computation has gone on for finitely many steps, the string consists of a $\triangleright$, possibly followed by a string of $\sqcup$s (y could be empty). Then string y is the *output of computation*, and denoted by $M(x) = y$.

For an integer $k \geq 1$, a *k-string (a multistring) Turing machine M^k* is defined as follows.

Definition 8 (k-string Turing machine)**.** *M^k is a quadruple (K, Σ, δ, s), where K, Σ and s are exactly as in ordinary Turing machine. M^k has multiple (k) strings. In M^k, δ is a function from $K \times \Sigma^k$ to $(K \cup \{h, \text{"yes"}, \text{"no"}\}) \times (\Sigma \times \{\leftarrow, \rightarrow, -\})^k$, denoted by $\delta(q, \sigma_1, \ldots, \sigma_k) = (p, \rho_1, D_1, \ldots, \rho_k, D_k)$. That is, δ decides the next state as before, but also decides for each of string the symbol overwritten. δ also decides the direction of cursor motion by looking at the current state and the current symbol at each string.*

In M^k, $\triangleright$ still cannot be overwritten or passed on to the left, that is, if $\sigma_i = \triangleright$, then $\rho_i = \triangleright$ and $D_i = \rightarrow$. Initially, all strings start with a $\triangleright$. The first string contains the input. The output of the computation of a k-string Turing machine M^k on input x is as with ordinary machines M, but the output can be read from the *last* kth string when the machine halts. In the following, a k-string Turing machine is denoted by M instead of M^k.

A *configuration* of M is a $(2k+1)$-tuple $(q, w_1, u_1, \ldots, w_k, u_k)$. Here $q \in K$ is a current state. $w_i u_i$ are strings in Σ^*. w_i is the ith string to the left of the ith cursor, including the symbol on the cursor, and u is the ith string to the right of the ith cursor (possibly empty). A configuration $(q, w_1, u_1, \ldots, w_k, u_k)$ *yields* a configuration $(q', q, w'_1, u'_1, \ldots, w'_k, u'_k)$ if $t(>0)$ steps of M from $(q, w_1, u_1, \ldots, w_k, u_k)$ results in configuration $(q', w'_1, u'_1, \ldots, w'_k, u'_k)$ (it is denoted by $(q, w_1, u_1, \ldots, w_k, u_k) \stackrel{M^t}{\to} (q', w'_1, u'_1, \ldots, w'_k, u'_k)$). If $(q, w_1, u_1, \ldots, w_k, u_k)$ yields $(q', w'_1, u'_1, \ldots, w'_k, u'_k)$ it is denoted by $(q, w_1, u_1, \ldots, w_k, u_k) \stackrel{M^*}{\to} (q', w'_1, u'_1, \ldots, w'_k, u'_k)$

Definition 9 (Nondeterministic Turing machine)**.** *A nondeterministic Turing machine is a quadruple $N = (K, \Sigma, \Delta, s)$. K, Σ and s are as a Turing machine M. Here Δ is a relation such that* $\Delta \subset (K \times \Sigma) \times [(K \cup \{h, \text{“yes”}, \text{“no”}\} \times \{\leftarrow, \rightarrow, -\})]$.

2.3 Languages and complexity classes

Let $L \subset (\Sigma \setminus \{\sqcup\})^*$ be a language. Let M be a multistring Turing machine such that, for any string $x \in (\Sigma \setminus \sqcup)^*$, if $x \in L$ then $M(x) = \text{“yes”}$ and if $x \notin L$ then $M(x) = \text{“no”}$. Then it is said that M *decides* L.

Let M be a k-string Turing machine and x be its input. If

$$(s, \triangleright, x, \triangleright, \epsilon, \ldots, \triangleright, \epsilon) \stackrel{M^t}{\to} (H, w_1, u_1, \ldots, w_k, u_k)$$

holds for some $H \in \{h, \text{“yes”}, \text{“no”}\}$, then the *time required by M on input x* is t. That is, the time required by M on x is the number of steps to halting. Let f be a function from the nonnegative integers to the nonnegative integers. If the time required by M on x is at most $f(|x|)$ for any input string x, then it is said that M *operates within time* $f(|x|)$. Also it is said that $f(|x|)$ is a *time bound* for M.

Definition 10 (DTIME). *Let $L \subset (\Sigma \setminus \{\sqcup\})^*$ be a language. If there is a multistring Turing machine operating in time $f(n)$ to decide L, then a set of all L is denoted by* $\mathbf{DTIME}(f(n))$.

$\mathbf{DTIME}(f(n))$ contains exactly those languages that can be decided by Turing machines with multiple strings operating within the time bound $f(n)$.

Let N be a nondeterministic Turing machine and x be its input. It is said that N

decides L,

$$(s, \triangleright, x) \xrightarrow{N^*} (\text{"yes"}, w, u)$$

holds for some strings w and u if and only if $x \in L$. Let f be a function from the nonnegative integers to the nonnegative integers.

Definition 11 (**NTIME**)**.** *If there is a nondeterministic Turing machine operating in time* $f(n)$ *to decide* L*, then a set of all* L *is denoted by* $\mathbf{NTIME}(f(n))$*.*

Let L be a language decided by some multistring Turing machine within a time polynomial of the length of the input x. That is, there is an integer $k > 0$ such that $L \in \mathbf{DTIME}(n^k)$.

Definition 12 (**P**)**.** *The union of* $\mathbf{DTIME}(n^k)$ *for all* $k > 0$ *is denoted by* $\mathbf{P}$*.*

Definition 13 (**NP**)**.** *The union of* $\mathbf{NTIME}(n^k)$ *for all* $k > 0$ *is denoted by* $\mathbf{NP}$*.*

P and **NP** are two of the most significant *complexity classes*. It is obvious that $\mathbf{P} \subseteq \mathbf{NP}$. The **P** versus **NP** problem, that is, whether $\mathbf{P} = \mathbf{NP}$ or not, is a major unsolved problem in computer science.

2.4 Graph Theory

In order to make this thesis self-contained, terms in graph theory are defined in this section. Graph-related problems addressed in this thesis are also defined.

2.4.1 Definitions and basic notations

Definition 14 (Undirected edge, Directed edge)**.** *An edge $e = (u, v)$ is a set with a relation of incidence that associates with each edge two vertices u and v. An edge is undirected if there is no distinction between (u, v) and (v, u). Otherwise (u, v) is considered to be directed edge from u to v.*

If there is an edge $e = (u, v)$ or $e = (v, u)$ then u is *adjacent* to v. u and v are *endpoints* of e.

Definition 15 (Multiple edges)**.** *Multiple edges (Parallel edges or a Multi-edge) are two or more distinct edges that are incident to the same two vertices u and v.*

Definition 16 (Graph)**.** *A graph $G = (V, E)$ is a pair of two sets, a set of vertices V and a set of edges E.*

$V(G)$ and $E(G)$ are a set of vertices and a set of edges of $G = (V, E)$ respectively. u and v are *independent* if neither (u, v) nor (v, u) exists in G. A set of vertices $I \subseteq V(G)$ is *independent* if for any pair (u, v) in I, u and v are independent.

Definition 17 (Loop)**.** *If $e = (v, v)$ then e is called a loop.*

Definition 18 (Simple graph)**.** *A graph that has no loop and no multiple edges is called a simple graph.*

Definition 19 (Undirected graph, Directed graph)**.** *If E is a set of undirected edges then $G = (V, E)$ is an undirected graph, otherwise a digraph or a directed graph.*

Definition 20 (Adjacency matrix)**.** *Let $G = (V, E)$ be an undirected graph with $V = \{v_1, \ldots, v_n\}$. The adjacency matrix $M_G = (m_{ij})$ of G is an $n \times n$ matrix such that m_{ij} is the number of edges whose endpoints are v_i and v_j.*

M_G is symmetric. If G is a simple graph then m_{ij} are either 0 or 1 for $i \neq j$ and $m_{ii} = 0$.

Throughout in this thesis, graphs are undirected and simple graphs unless otherwise noticed.

Definition 21 (Singleton)**.** *For a vertex $v \in V(E)$, if there is no vertex $u(\neq v)$ such that $e = (v, u) \in E(G)$ then v is called a singleton or an isolated vertex in G.*

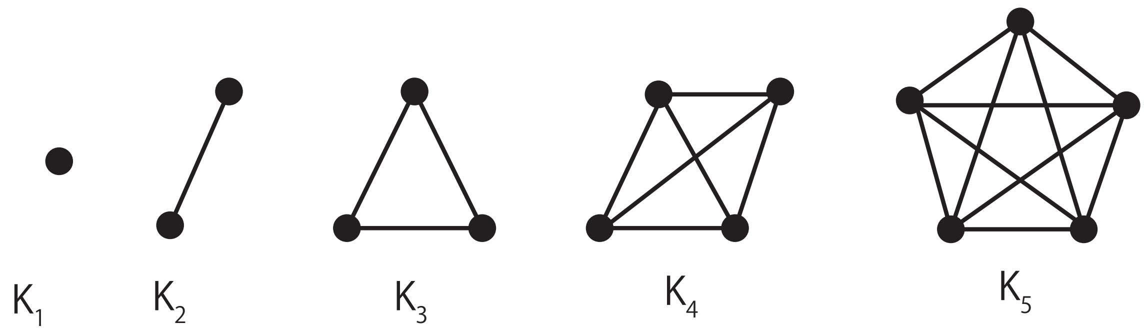

Figure 2.1: Complete graphs K_n $(n = 1, \dots, 5)$.

Definition 22 (Complement graph)**.** *A graph $\bar{G}$ is the complement graph of G such that $V(G') = V(G)$ and $E(G') = [V(G)]^2 \backslash E(G)$.*

Definition 23 (Neighborhood)**.** *The set of all vertices adjacent to v is the neighborhood of v denoted by $N_G(v)$ or $\Gamma(v)$.*

Definition 24 (Degree)**.** *The number of all edges incident to v is the degree of v denoted by $d_G(v)$.*

Definition 25 (Complete graph)**.** *A graph G is a complete graph if for all $u, v \in V$ there is an edge $(u, v) \in E$.*

A complete graph is denoted by K_n where $n = |V|$. See Fig. 2.1.

Definition 26 (Subgraph)**.** *G' is a subgraph of G if $E(G') \subseteq E(G)$ and $V(G') \subseteq V(G)$. Then G is a supergraph of G'.*

Definition 27 (Induced subgraph)**.** *Let G' be a subgraph of G. If $(u, v) \in E(G')$ holds for all $u, v \in V(G')$ such that $(u, v) \in E(G)$, then G' is an induced subgraph of G. The subgraph induced by a set of vertices W is denoted by $G[W]$.*

Definition 28 (Path)**.** *A path $P_k = (V, E)$ is a graph with distinct vertices $V = \{v_0, v_1, \dots, v_k\}$ and edges $E = \{(v_0, v_1), (v_1, v_2), \dots, (v_{k-1}, v_k)\}$.*

Definition 29 (Length)**.** *The length of a path P_k is the number of edges $k = |E(P)|$.*

Definition 30 (Distance)**.** *The distance between two vertices u and v in G is the shortest length of paths that have u and v.*

Definition 31 (Chord)**.** *Let P_k be a path. An edge (v_i, v_j) is a chord of P_k if $|i - j| > 1$.*

Definition 32 (Connected graph)**.** *A connected graph G is a graph such that for any $u, v \in V(G)$ there is a path P_k such that $u, v \in V(P_k)$.*

Definition 33 (Distance hereditary graph)**.** *A distance hereditary graph is a graph G in which the distances in any connected induced subgraph are the same as they are in G.*

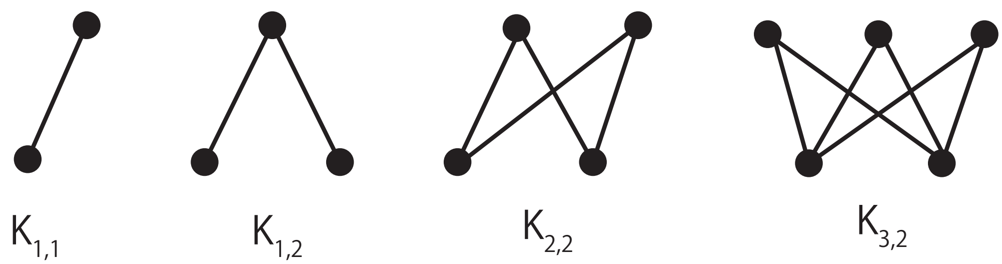

Figure 2.2: Complete bipartite graphs $K_{1,1}$, $K_{1,2}$, $K_{2,2}$ and $K_{3,2}$.

Definition 34 (Cycle). *For $k \geq 3$, a cycle $C_k = (V, E)$ is a path with vertices $V = \{v_0, v_1, \ldots, v_k\}$ and edges $E = \{(v_0, v_1), (v_1, v_2), \ldots, (v_{k-1}, v_0)\}$.*

If k is odd then C_k is an *odd cycle*, otherwise an *even cycle*.

Definition 35 (Clique). *A clique of G is an induced subgraph of G that is a complete graph.*

Definition 36 (Bipartite graph). *A bipartite graph $B = (X, Y, E)$ is a graph $G(V, E)$ such that $V = X \cup Y$, $X \cap Y = \emptyset$, and $E \subseteq X \times Y$.*

Note that this notation specifies the partition of V into X and Y, also a bipartite graph has no odd cycle as a subgraph.

Definition 37 (Complete bipartite graph). *A complete bipartite graph is a bipartite graph $B = (X, Y, E)$ such that there is an edge $(u, v) \in E$ for all $u \in X$ and $v \in Y$.*

A complete bipartite graph is denoted by $K_{m,n}$ where $m = |X|$ and $n = |Y|$. See Fig. 2.2.

Definition 38 (Stargraph). *A star or stargraph is a complete bipartite graph $K_{m,n}$ such that either $m = 1$ or $n = 1$.*

Definition 39 (Biclique). *A biclique of G is a subgraph of G that is a complete bipartite graph.*

Note that if G is a bipartite graph then a biclique of G is an induced subgraph of G.

Definition 40 (Bisimplicial edge). *A bismplicial edge $e = (u, v)$ is an edge in G such that the induced subgraph $G[N_G(v) \cup N_G(u)]$ is a biclique $(N_G(v), N_G(u), E)$ of G.*

Definition 41 (Maximal biclique). *A biclique K in G is a maximal biclique if there is no biclique K' in G such that $V(K) \subset V(K')$ and $E(K) \subset E(K')$.*

Definition 42 (Domino). *A domino is a graph with vertices $V = \{x_1, x_2, x_3, y_1, y_2, y_3\}$ and edges $E = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_1, y_2), (x_2, y_1), (x_2, y_3), (x_3, y_2)\}$.*

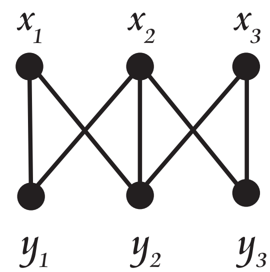

Figure 2.3: A domino.

See Fig. 2.3.
Definition 43 (Domino-free graph, C_4-free graph)**.** *A domino-free graph is a graph that has no domino as an induced subgraph. A C_4-free graph is a graph that has no C_4 as an induced subgraph.*

2.4.2 Graph-related problems

In this section, graph-related problems mentioned in this thesis are defined.

Let $C \subseteq V$ be a clique of $G = (V, E)$, and $I \subseteq V$ is an independent set in G.
Definition 44 (The maximum independent set problem)**.** *The maximum independent set problem is the problem of finding a set of independent vertices such that the number of the vertices is maximum.*

Let $\alpha(G) = \max\{|I| : I \text{ is an independent set in } G\}$.
Definition 45 (The maximum clique problem (MAX CLIQUE))**.** *The maximum clique problem is the problem of finding a clique with the maximum size.*

Let $\omega(G) = \max\{|C| : C \text{ is a clique in } G\}$.

A *vertex cover* S of a graph G is a set of vertices of G such that every edge of G has at least one of the member of S as an endpoint.
Definition 46 (The vertex cover problem (VERTEX COVER))**.** *The vertex cover problem is the problem of finding a vertex cover with the minimum size.*

Let $\sigma(G) = \min\{|S| : S \text{ is a vertex cover of } G\}$.

Let $\mathcal{S} = \{s_1, \ldots, s_k\}$ be a set of subgraphs of G. If $\cup_{i=1}^{k} V(s_i)$ is a vertex cover of G then G is *vertex-covered (or covered)* by the set of graphs $\mathcal{S}$.

Definition 47 (The independent set cover problem)**.** *The independent set cover problem is the problem of finding a set of independent sets $\mathcal{S} = \{I_1, \ldots, I_k\}$ of G such that $\mathcal{S}$ covers G and k is the minimum.*

Let $\chi(G) = \min\{k : \cup_{i=1}^{k} I_i \text{ is a vertex cover of } G\}$

Definition 48 (The clique cover problem)**.** *The clique cover problem is the problem of finding a set of cliques $\mathcal{S} = \{C_1, \ldots, C_k\}$ of G such that $\mathcal{S}$ covers G and k is the minimum.*

Let $\kappa(G) = \min\{k : V \text{ is covered by a set of cliques } C_1, C_2, \ldots, C_k \text{ in } G\}$

Definition 49 (The biclique cover problem)**.** *The biclique cover problem is the problem of finding a set of bicliques $\mathcal{B} = \{B_1, \ldots, B_k\}$ of G such that $\mathcal{B}$ covers G and k is the minimum.*

Let $\kappa_B(G) = \min\{k : V \text{ is covered by a set of bicliques } B_1, B_2, \ldots, B_k \text{ in } G\}$.

Let $\mathcal{B}$ be a set of bicliques of G. If the union of all edges of all member of $\mathcal{B}$ is $E(G)$, then $E(G)$ (or G) is *edge-covered* by $\mathcal{B}$.

Definition 50 (The biclique edge cover problem (BEC))**.** *The biclique edge cover problem is the problem of finding a set of bicliques $\mathcal{B} = \{B_1, \ldots, B_k\}$ of G such that $\mathcal{B}$ edge-covers G and k is the minimum.*

Let $c(G) = \min\{k : E \text{ is edge - covered by a set of bicliques } B_1, B_2, \ldots, B_k \text{ in } G\}$.

Let $\mathcal{B} = \{B_1, B_2, \ldots, B_k\}$ be a set of bicliques of G such that $E(B_i) \cap E(B_j) = \emptyset$ for $i \neq j$. If the union of all edges of all member of $\mathcal{B}$ is $E(G)$, then $E(G)$ (or G) is *edge-partitioned* by $\mathcal{B}$.

Definition 51 (The biclique edge partition problem (BEP))**.** *The biclique edge partition problem is the problem of finding a set of bicliques $\mathcal{B} = \{B_1, \ldots, B_k\}$ of G such that $\mathcal{B}$ edge-partitions G and k is the minimum.*

Let $b(G) = \min\{k : E \text{ is edge - partitioned by a set of bicliques } B_1, B_2, \ldots, B_k \text{ in } G\}$

Let each vertex in $V(G)$ be assigned an alphabet (symbol or color). Let k be a number of alphabets that are assigned to $V(G)$. If there is an assignment such that two of vertices x and y are assigned different alphabets for each $e = (x, y) \in E(G)$, then G is *k-colorable.*

Definition 52 (The chromatic number problem)**.** *The chromatic number problem is the problem of finding $\chi_C(G)$ such that*

$$\chi_C(G) = \min\{k : G \text{ is } k\text{ - } colorable\}.$$

$\chi_C(G)$ is called the chromatic number of G.

2.5 Satisfiability

The constraint satisfaction problem (CSP) and the satisfiability problem (SAT) have played central role in the computational complexity theory. In this section, these problems are defined.

2.5.1 CSP: Constraint satisfaction problem

Let $V = \{v_i, \ldots, v_n\}$ be a set of variables on a finite alphabet Σ.

Definition 53 (Constraint)**.** *A q-ary constraint $(C, i_1, \ldots, i_q)$ consists of a subset $C \subseteq \Sigma^q$ and a q-tuple of indices of variables $i_1, \ldots, i_q \in \{1, \ldots, n\}$. An assignment a is a mapping $a : V \to \Sigma$. A constraint is satisfied by a given assignment a if and only if $(a(v_{i_1}), \ldots, a(v_{i_q})) \in C$.*

Definition 54 (CSP: Constraint satisfaction problem)**.** *Let $\mathcal{C} = \{C_1, \ldots, C_N\}$ be a set of q-ary constraints C_i over a set of variables V. The constraint satisfaction problem (CSP) on $\mathcal{C}$ is the problem deciding whether there is an assignment of the variables that satisfies every constraint.*

Definition 55 (UNSAT($\mathcal{C}$))**.** UNSAT($\mathcal{C}$) *is the smallest fraction of unsatisfied constraints over all possible assignments for V.*

Note that $\mathcal{C}$ is satisfiable if and only if UNSAT($\mathcal{C}$) $= 0$.

2.5.2 SAT: Satisfiability problem

The satisfiability problem(SAT) is the special case of CSP such that constraints are Boolean expressions and variables are over the truth values. Let $X = \{x_1, x_2, \ldots\}$ be a countably infinite alphabet of Boolean variables. These variables can take the two truth values *TRUE* and *FALSE* (or 1 and 0). These variables together with the *Boolean connectives* $\vee$ (logical or), $\wedge$ (logical and), and $\neg$ (logical not), form *Boolean expressions* as follows.

Definition 56 (Boolean expression)**.** *A Boolean expression can be any one of (a) a Boolean variable, such as x_i, or (b) an expression of the form $\neg\phi_1$, where ϕ_1 is a Boolean expression, or (c) an expression of the form $(\phi_1 \vee \phi_2)$, where ϕ_1 and ϕ_2 are Boolean expressions or (d) an expression of the form $(\phi_1 \wedge \phi_2)$, where ϕ_1 and ϕ_2 are Boolean expressions. In case (b) the expression is called the negation of ϕ_1; in case (c), it is the disjunction of ϕ_1 and ϕ_2; in case (d), it is the conjunction of ϕ_1 and ϕ_2. An expression of the form x_i or $\neg x_i$ is called a literal.*

Definition 57 (Truth assignment)**.** *A truth assignment T is a mapping from a finite set X' of Boolean variables, $X' \subset X$, to the set of truth values TRUE and FALSE.*

A Boolean expression ϕ is *satisfied* by an assignment a such that ϕ be TRUE.

The subset of X, $X(\phi)$ is *the Boolean variables occurring (appearing) in* ϕ defined as follows. If ϕ is a Boolean variable x_i, then $X(\phi) = \{x_i\}$. If $\phi = \neg\phi_1$, then $X(\phi) = X(\phi_1)$. If $\phi = (\phi_1 \vee \phi_2)$, or if $\phi = (\phi_1 \wedge \phi_2)$, then $X(\phi) = X(\phi_1) \cup X(\phi_2)$.

Definition 58 (CNF: Conjunctive normal form)**.** *A Boolean expression ϕ is in conjunctive normal form (CNF) if $\phi = \bigwedge_{i=1}^{n} C_i$ where each of the C_is is the disjunction of one or more literals. C_i is a clause of ϕ. k-CNF is a CNF such that each C_i has at most k literals.*

Definition 59 (SAT: Satisfiability problem)**.** *Let ϕ be a Boolean expression ϕ in CNF. The satisfiability problem (SAT) for ϕ is the problem of deciding whether there is an assignment of the variables that satisfies ϕ .*

SAT is an NP-complete problem [Coo71]. k-SAT is SAT on k-CNF ϕ.

Definition 60 (Ek-SAT)**.** *Ek-SAT is SAT on CNF ϕ in which each clause has exactly k literals.*

Definition 61 (k'-OCC-Ek-SAT)**.** *k'-OCC-Ek-SAT is Ek-SAT on CNF ϕ in which every variables occurs exactly k' times.*

Definition 62 (MAX SAT)**.** *MAX SAT for ϕ is the problem to find the largest fraction of satisfied clauses of ϕ over all possible assignments.*

Definition 63 (MAX Ek-SAT)**.** *MAX Ek-SAT for ϕ is the problem to find the largest fraction of satisfied clauses of ϕ in which each clause has exactly k literals over all possible assignments.*

Definition 64 (k'-OCC-MAX Ek-SAT)**.** *k'-OCC-MAX Ek-SAT for ϕ is the problem to find the largest fraction of satisfied clauses of ϕ in which each clause has exactly k literals and every variables occurs exactly k' times over all possible assignments.*

2.6 NP-completeness

A *decision problem* is to ask whether a given input satisfies a certain property. Assume that x is a string $x \in \{0,1\}^*$ encoded the input of a decision problem A. Assume that there is a multistring Turing machine M such that M outputs "YES" if x satisfies problem's property, otherwise outputs "No". That is, the set of x that outputs "YES" is a language L decided by M. Then the decision problem A is in $\mathbf{P}$, if L is decided by M within a time polynomial of $|x|$. For a decision problem B, assume that there is a nondeterministic Turing machine N that outputs "YES" if and only if the input of the decision problem B satisfies problem's property. Let L be the set of inputs that output "YES". Then the decision problem B is in $\mathbf{NP}$, if L is decided by N within a time polynomial of the size of the input of the problem. In the following, a decision problem is identified to a language that encodes the input of the problem.

Let A and B be two decision problems. If there is a polynomial time computable function f such that $x \in A$ if and only if $f(x) \in B$, then it is said that A *reduces* B. It is denoted by $A \leq B$. If $A \leq B$ and $B \in \mathbf{P}$, then $A \in \mathbf{P}$ holds. If $A \leq B$ and $B \leq C$ then $A \leq C$ holds.

Definition 65 ($\mathbf{NP}$-hard)**.** *If $L \leq A$ holds for every problem $L \in \mathbf{NP}$, then a decision problem A is $\mathbf{NP}$-hard.*

Definition 66 ($\mathbf{NP}$-complete)**.** *A is $\mathbf{NP}$-complete if A is $\mathbf{NP}$-hard and it belongs to $\mathbf{NP}$.*

We denote decision version of the optimization problems defined in Section 2.4.2 as follows. INDEPENDENT SET(k) is the decision problem such that for given a graph G deciding whether $\alpha(G) \geq k$ holds or not. VERTEX COVER(k) is the decision problem such that for given a graph G deciding whether $\sigma(G) \leq k$ holds or not. For the other problems we denote the decision version of problems in the same way. CHROMATIC NUMBER(k) is also denoted by k-COLORABLITY.

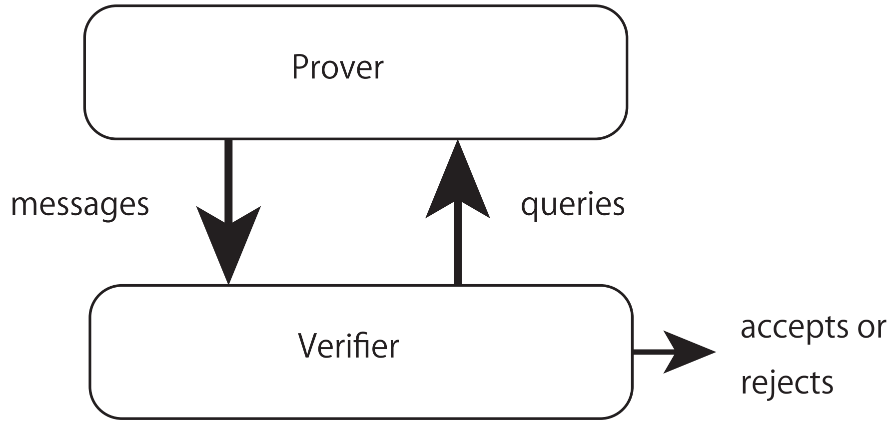

Figure 2.4: Verifier and Prover.

2.7 Interactive proof system

The Church-Turing thesis states that a function is *algorithmically computable* if and only if it is computable by a Turing machine. That is, if some method (*algorithm*) exists to carry out a calculation, then the same calculation can also be carried out by a Turing machine. In the following, a multistring Turing machine M that decides L is identified to an algorithm $V(x, y)$ such that it decides whether there is a relation (x, y) or not.

We defined the complexity class **NP** in Definition 13. There is another characterization of **NP** as follows. A language L is in **NP**, if and only if there are an algorithm $V(\cdot, \cdot)$ running in polynomial time and a polynomial $p(\cdot)$ such that

$$x \in L \Leftrightarrow \exists y,\ |y| \leq p(|x|) \,\text{and}\, V(x, y) \,\text{accepts.}$$

That is,

$$\begin{aligned} x \in L &\Rightarrow \exists y, |y| \leq p(|x|) \,\text{and}\, V(x, y) \,\text{accepts (Completeness)} \\ x \notin L &\Rightarrow \forall y, |y| \leq p(|x|) \,\text{and}\, V(x, y) \,\text{rejects (Soundness).} \end{aligned}$$

Also **NP** can be characterized by using the *interactive proof* with a *Prover* P_o and a *Verifier* V_e. *Prover* P_o can do computations with unbounded time and space. *Verifier* V_e asks *queries* to P_o, and runs in polynomial time with messages from P_o. The task of the prover is to convince the verifier that $x \in L$. Here, a sequence of queries and messages is exchanged between P_o and V_e. *Strategy* for P_o is a function from the sequence of messages that P_o send to V_e so far (thus seen by V_e) to the next message. V_e is *convinced* if and only if V_e can verify that $x \in L$ using the sequence of messages from P_o.

Now the class **NP** is the set of languages L such that

$$\begin{aligned} x \in L &\Rightarrow P_o \text{ has a strategy to convince } V_e \text{ (Completeness)} \\ x \notin L &\Rightarrow P_o \text{ has no strategy to convince } V_e \text{ (Soundness),} \end{aligned}$$

where V_e may be a randomized machine, and may ask further questions of the prover based on the messages that have been set to it. Exchanging messages are called *interactions*.

By adding randomness to the class **NP**, the class **MA** is defied as follows.
Definition 67 (**MA**)**.** $L \in \mathbf{MA}$ *if and only if there exists a probabilistic algorithm* V_r *such that:*

$$\begin{aligned} x \in L &\Rightarrow \exists y,\ \Pr[V_r(x, y) \text{ accepts}] \geq \frac{2}{3} \\ x \notin L &\Rightarrow \forall y,\ \Pr[V_r(x, y) \text{ accepts}] \leq \frac{1}{3}. \end{aligned}$$

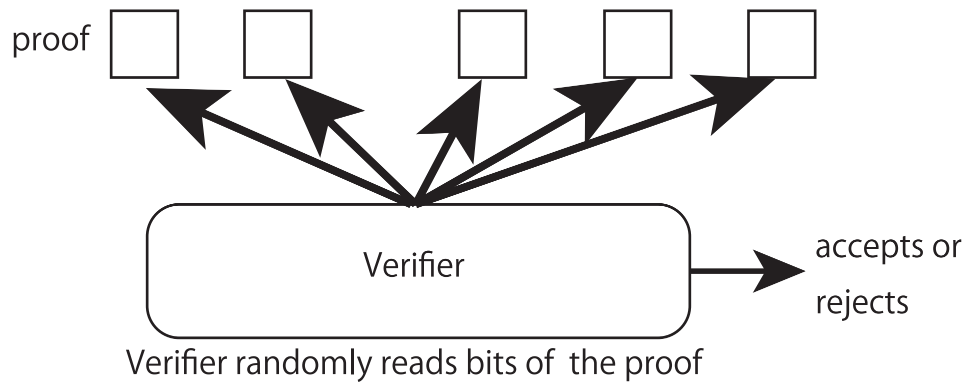

Figure 2.5: The probabilistically checkable proof.

2.8 PCP:Probabilistically checkable proof

Let x be a string with a length $n = |x|$. The problem to decide whether $x \in L$ or not, is in the complexity class $\mathrm{PCP}_{c(n),s(n)}(r(n), q(n))$ defined as follows.

Definition 68 ($\mathrm{PCP}_{c(n),s(n)}(r(n), q(n))$)**.** *The decision problem is in* $\mathrm{PCP}_{c(n),s(n)}(r(n), q(n))$, *if there is* V_e *such that*

$$\begin{aligned} x \in L &\;\; \Rightarrow \;\; \exists\pi,\ \Pr[V_e \text{ accepts}] \geq c(n) \\ x \notin L &\;\; \Rightarrow \;\; \forall\pi,\ \Pr[V_e \text{ accepts}] \leq s(n) \end{aligned}$$

using $O(r(n))$ *of random bits and read* $O(q(n))$ *bits of the proof* π*.*

Theorem 1 (The PCP theorem)**.** [AS98] $NP = PCP_{1,1/2}(\log n, 1)$.

In the early 1990s, the PCP theorem was proved with a little too complex transformations [AS98]. A recent work due to Dinur[Din07] gives a simple construction of probabilistically checkable proofs. We briefly introduce the proof of the PCP theorem following [Din07].

Definition 69 (Constraint graph)**.** $\mathcal{G} = \langle (V, E), \Sigma, \mathcal{C} \rangle$ *is a constraint graph, if*

1. (V, E) *is an undirected graph, called the underlying graph of* $\mathcal{G}$*.*
2. *The set* V *is also viewed as a set of variables such that each of them has a value over alphabet* Σ*.*
3. *Each edge* $e \in E$ *has a constraint* $c(e) \subseteq \Sigma \times \Sigma$*, and* $\mathcal{C} = \{c(e) \mid e \in E\}$*. A constraint* $c(e)$ *is said to be satisfied by* (a, b) *if and only if* $(a, b) \in c(e)$*.*

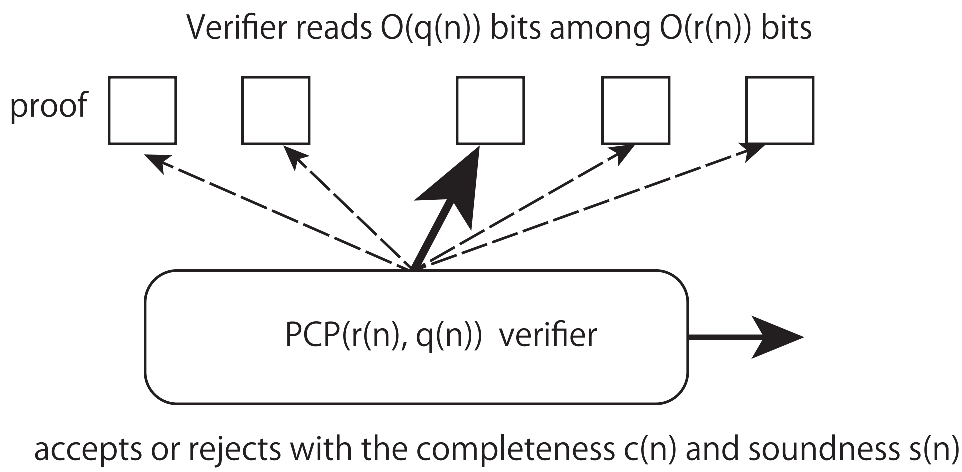

Figure 2.6: $\mathrm{PCP}_{c(n),s(n)}(r(n), q(n))$.

Note that if $c(e) = \{(1,2),(1,3),(2,1),(2,3),(3,1),(3,2)\}$ for all e, then the problem to decide whether $\mathcal{C}$ is satisfiable or not is 3-COLORABILITY.

Let a mapping $\sigma : V \to \Sigma$ be an assignment to vertices in V. For any assignment σ, $\mathrm{UNSAT}_\sigma(\mathcal{G})$ and $\mathrm{UNSAT}(\mathcal{G})$ are defined as follows.
Definition 70 (UNSAT($\mathcal{G}$))**.**

$$\begin{aligned}\mathrm{UNSAT}_\sigma(\mathcal{G}) &= \prod_{(u,v)\in E} \Pr[(\sigma(u),\sigma(v)) \notin c(e)] \\ \mathrm{UNSAT}(\mathcal{G}) &= \min_\sigma \mathrm{UNSAT}_\sigma(\mathcal{G})\end{aligned}$$

3-COLORABILITY is **NP**-hard, thus it is **NP**-hard to distinguish between the cases $\mathrm{UNSAT}(\mathcal{G}) = 0$ and $\mathrm{UNSAT}(\mathcal{G}) \neq 0$.

Let $size(\mathcal{G}) = |V| + |E|$. Dinur[Din07] proved the following theorem.
Theorem 2 (The gap amplification)**.** *There exist Σ_0 such that the following holds. For any finite alphabet Σ there exist constants $C > 0$ and $0 < \alpha < 1$ such that, given a constraint graph $\mathcal{G} = \langle (V,E), \Sigma, \mathcal{C}\rangle$, one can construct, in polynomial time, a constraint graph $\mathcal{G}' = \langle (V',E'), \Sigma_0, \mathcal{C}'\rangle$ such that*

- $size(\mathcal{G}') \leq C \cdot size(\mathcal{G})$.
- *If* $\mathrm{UNSAT}(\mathcal{G}) = 0$ *then* $\mathrm{UNSAT}(\mathcal{G}') = 0$ *(Completeness).*
- $\mathrm{UNSAT}(\mathcal{G}') \geq \min(2 \cdot \mathrm{UNSAT}(\mathcal{G}), \alpha)$ *(Soundness).*

If $\mathrm{UNSAT}(\mathcal{G}') \neq 0$, by repeating this construction, we have the constraint graph $\mathcal{G}_{\mathrm{final}}$ such that $\mathrm{UNSAT}(\mathcal{G}_{\mathrm{final}}) \geq \alpha$ holds. Let $\mathcal{G}'_{\mathrm{final}}$ be the set of constraint graphs of copied $\mathcal{G}_{\mathrm{final}}$ for k times. Then the probability that each copy of $\mathcal{G}_{\mathrm{final}}$ in $\mathcal{G}'_{\mathrm{final}}$ is satisfied by k distinct assignments, is at most $(1-\alpha)^k$. If $k \geq -\log_2(1-\alpha)$ then $\mathrm{UNSAT}(\mathcal{G}'_{\mathrm{final}}) \geq 1/2$. As we can construct such a system of constraint graphs from an instance of 3-COLORABILITY in polynomial time, the next theorem follows from Theorem 2.

Theorem 3 (Inapproximability version of the PCP theorem)**.** *There are integers $q > 1$ and $|\Sigma| > 1$ such that, given an input a collection $\mathcal{C}$ of q-ary constraint over an alphabet Σ, it is* **NP***-hard to decide whether* $\mathrm{UNSAT}(\mathcal{C}) = 0$ *or* $\mathrm{UNSAT}(\mathcal{C}) \geq 1/2$.

The proof of Theorem 1 is as follows.

Proof. ($L \in NP \Rightarrow L \in \mathrm{PCP}_{1,1/2}(\log n, 1)$) We show that every **NP** language has a verification procedure *Ver* that reads $c \log n$ random bits, accesses $q = O(1)$ bits from the proof and decides whether to accept or reject (where q and c are constants). For each fixed random bit pattern $r \in \{0,1\}^{c \log n}$, *Ver* deterministically reads a fixed set of q bits from the proof: $i_1^{(r)}, \ldots, i_q^{(r)}$. Denote by $C^{(r)} \subseteq \{0,1\}^{c \log n}$ the possible contents of the accessed proof bits that would cause *Ver* to accept.

We present a reduction from L to gap constraint satisfaction. Let $\Sigma = \{0,1\}$ and $x \in \Sigma^n$ be the input. Put a Boolean variable for each proof location accessed by *Ver* on input x. The length of string of these Boolean variables is at most $q2^{c \log n} = qn^c$. Let $r_0 = \{c \log n\}$. Construct a system of constrains $\mathcal{C}_x = \{c_r\}_{r \in \{0,1\}^{r_0}}$ such that the constraint c_r is defined by $c_r = \{C^{(r)}, i_1^{(r)}, \ldots, i_q^{(r)}\}$. We observe that the rejection probability of *Ver* is exactly equal $\mathrm{UNSAT}(\mathcal{C}_x)$ so it is zero if $x \in L$, and at least $1/2$ if $\notin L$.

($L \in NP \Leftarrow L \in \mathrm{PCP}_{1,1/2}(\log n, 1)$) For the converse, assume there is a reduction taking instances of any **NP**-language into constraint systems such that the gap property holds. Here is how to construct a verifier. The verifier will first (deterministically) compute the constraint system output by the reduction guaranteed above. It will expect the proof to consist of an assignment for the variables of the constraint system. Next, the verifier will use its random string to select a constraint uniformly at random, and check that the assignment satisfies it, by querying the proof at the appropriate locations. □

We omit parameters $c(n)(= 1)$ and $s(n)(= 1/2)$ in the following, unless otherwise noticed.

2.9 Approximation algorithms

Both of the bicliqe edge partition and the biclique edge cover problems are NP-hard. Thus we cannot have exact solutions for them in polynomioal time.

Let I be an instance of an optimization problem. A *solution* $S(I)$ is an object that satisfies some specified property required by the problem. A solution $S(I)$ is associated with a value of an objective function. $|S(I)|$, the *size* of $S(I)$, is the value of the objective function.

The hardness of approximation is measured by an *approximation ratio* ρ. A ρ-approximation algorithm for a minimization (maximization) problem is the algorithm that guarantees to return a solution with the size at most ρ (at least $1/\rho$) times the size of the optimal solution.

Definition 71 (Approximation algorithm)**.** *Let P' be an* NP*-hard optimization problem. Let I be an instance of P', and $S_o(I)$ be an optimal solution. If P' is a minimization problem, then a ρ-approximation algorithm is an algorithm that gives a solution $S(I)$ for any instance I such that*

$$\rho|S_o(I)| \geq |S(I)|$$

in polynomial time. If P' is a maximization problem, then a ρ-approximation algorithm is an algorithm that gives a solution $S(I)$ for any instance I such that

$$|S_o(I)|/\rho \leq |S(I)|$$

in polynomial time.

Note that $\rho > 1$ holds for both of minimization and maximization NP-hard problems. The factor ρ is called the approximation ratio (or the relative performance guarantee) of an approximation algorithm for an NP-hard problem.

For example, let us see an approximation algorithm for the minimum vertex cover problem. Let S_o be an optimal solution of the minimum vertex cover problem. Let E' be the set of edges (u, v) chosen in Algorithm 1. The set of vertices in S_o covers every edge in E'. Thus vertices in S_o include at least one of the endpoints of each edge in E'. As no two edges in E' share an endpoint, $|S_o| \geq |E'|$ holds. Algorithm 1 gives S such that $|S| = 2|E'|$. Thus this algorithm guarantees to give a solution with the size at most $2|S_o|$.

NP-hard problems that have a ρ-approximation algorithm with any constant $\rho > 1$ is not hard to solve in the sense of approximation. These problems are said to have a

Algorithm 1 The 2-approximation algorithm for the vertex cover problem.

Input a graph $G = (V, E)$
$S \Leftarrow \emptyset$
while $E(G) \neq \emptyset$ **do**
  choose arbitrarily an edge $(u, v) \in E(G)$
  $S \Leftarrow S \cup \{u, v\}$
  Delete u, v and their incident edges from G. Let G be the resulted graph.
end while
Output S

polynomial-time approximation scheme (PTAS). For example, the knapsack problem has PTAS. On the other hand, there are some problems that has no ρ-approximation algorithm with some constant ρ. For example, the minimum vertex cover problem cannot be approximated within a factor of 1.3606 for any sufficiently large vertex degree [DS05]. Furthermore, there are some problems that have no ρ-approximation algorithm with any constant ρ. For example, it is known that the set cover problem cannot be approximated within the ratio $(1 - O(1)) \ln n$ [Fei98]. Also, the maximum clique problem cannot be approximated within the ratio $O(n^{1-\epsilon})$ for any $\epsilon > 0$ [Hås99].

Definition 72 (PTAS: A polynomial time approximation scheme)**.** *A polynomial time approximation scheme (PTAS) is a family of algorithms $\{A_\rho\}$, where there is an algorithm for each $\rho > 1$, such that A_ρ is a ρ-approximation algorithm.*

MAX E3-SAT has no PTAS. However, it has an 8/7- approximation algorithm. Let us see this algorithm.

Let Φ be an instance of MAX E3-SAT with m clauses. Let x_i and ϕ_j be a variable and a clause of Φ, respectively. Suppose that x_i be a probabilistic variable that is either TRUE or FALSE with probabilty 1/2. Let $E[\phi]$ be the expected number of satisfied clauses in ϕ. Note that $E[\phi_j] = 7/8$ and $E[\Phi] = 7m/8$ hold. Let $\Phi_{x_i=t}$ be Φ such that x_i is fixed to $t \in \{\mathrm{TRUE}, \mathrm{FALSE}\}$. Then

$$E[\Phi] = \frac{1}{2}(E[\Phi_{x_1=\mathrm{TRUE}}] + E[\Phi_{x_1=\mathrm{FALSE}}])$$

holds. If $E[\Phi_{x_1=\mathrm{TRUE}}] > E[\Phi_{x_1=\mathrm{FALSE}}]$ then set $x_1 = \mathrm{TRUE}$ otherwise $x_1 = \mathrm{FALSE}$. Repeat this pocedure to Φ until values of all variables are fixed. It is easy to see that the expected number of satisfied clauses never decreased in each steps. Thus we have an assignment that satisfies at least $7m/8$ clauses of Φ.

Let Π be an NP-hard minimization problem. To obtain an approximation hardness result of Π, a reduction from an NP-complete decision problem Π_D into a set of instances of Π is constructed. In these reductions, if any "Yes" instance of Π_D is mapped to an instance of Π with objective function value $\leq k$, whereas any "No" instance of Π_D is mapped to an instance of Π with objective function value $\geq k+1$. This implies that obtaining an approximation algorithm with approximation ratio better than $(k+1)/k$ is not possible unless P = NP, since this would then allow us to distinguish between the "Yes" and "No" instances of NP-complete decision problem Π_D.

As mentioned in Chapter 1, although MAX 3-SAT has an 8/7-approximation algorithm [KZ97], it cannot be approximated within a ratio of $8/7 - \epsilon$ for any $\epsilon > 0$ [Hås01]. Furthermore, MAX CLIQUE has no ρ-approximation algorithm with a constant ρ. Let APX denote the class of approximation problems that have ρ-approximation algorithms. Then MAX 3-SAT belongs to APX, whereas MAX CLIQUE does not belong APX. These hardness results are mostly derived by reductions using the PCP theorem noticed in the following sections.

2.10 Hardness of MAX 3-SAT

MAX 3-SAT is the optimization version of 3-SAT. (See Definition59.) That is, MAX 3-SAT is the problem to find the maximum fraction of satisfied clauses in 3-CNF formula ϕ. As MAX 3-SAT is **NP**-hard, it cannot be solved in polynomial time unless P = NP. The next theorem follows from the PCP theorem.

Theorem 4. There is an $\epsilon > 0$ such that $(1-\epsilon)$-approximation of MAX 3-SAT is **NP**-hard.

Proof. Let L be a language such that the problem deciding whether $x \in L$ or not is **NP**-complete. Let $|x| = n$. By the PCP theorem, $L \in \mathrm{PCP}(\log n, 1)$ holds. Let V be the verifier for L. Then there is two constants c and q such that V can check whether $x \in L$ or not by using $c \cdot \log n$ random bits and reading q bits from the proof.

Given an instance x of L, we construct a 3-CNF formula ϕ_x on m clauses such that for some ϵ

$$\begin{aligned} x \in L &\Rightarrow \phi_x \text{ is satisfiable} \\ x \notin L &\Rightarrow \text{ no more than } (1-\epsilon)m \text{ clauses of } \phi_x \text{ are satisfiable.} \end{aligned}$$

Then this theorem holds from the observation of the previous section. Such ϕ_x can be constructed as follows. By using $c \cdot \log n$ random bits, V chooses one string r from $2^{c \cdot \log n} (= n^c)$ number of strings. For each r, V reads q bits from the proof. Let $i_1, \ldots, i_q$ be positions of the proof that V reads by choosing r. For each r, let $f_r : \{0,1\}^q \to \{0,1\}$ be a Boolean function such that V accepts if and only if $f_r(\pi(i_1), \ldots, \pi(i_q)) = 1$. Note that $\pi(i)$ is a value of the proof at the ith position. Now we have n^c functions f_r that have q variables. Each function f_r can be transformed to a CNF formula ϕ_r of 2^q clauses such that $\phi_r = \mathrm{TRUE}$ under π if and only if $f_r(\pi(i_1), \ldots, \pi(i_q)) = 1$. A clause with $k > 3$ literals can be transformed to clauses such that each clause has exact three literals as follows. Let $y_1, \ldots, y_{q-3}$ be $(q-2)$ auxiliary variables. A clause $c_r = (x_{i_1} \vee x_{i_2} \vee \cdots \vee x_{i_q})$ is transformed to $c'_r = (x_{i_1} \vee x_{i_2} \vee y_1) \wedge (\bar{y}_1 \vee x_{i_3} \vee y_2) \wedge (\bar{y}_2 \vee x_{i_4} \vee y_3) \wedge \cdots \wedge (\bar{y}_{q-3} \vee x_{i_{q-1}} \vee x_{i_q})$. Note that c'_r has $(q-2)$ clauses. Let ϕ'_r be the 3-CNF formula to which all clauses of ϕ_r is transformed by the above way. It is easy to check that if c_r is satisfiable then there is an assignment that satisfies all clauses of c'_r. If there is no assignment that satisfies c_r, then at least one clause of c'_r is unsatisfied by any assignment. If V reject x with probability more than $1/2$, more than $n^c/2$ of ϕ_R are unsatisfiable. Thus, in this case, more than $n^c/2$ of $n^c \cdot 2^q(q-2)$ clauses of $\bigwedge_r \phi'_r$ are unsatisfied by any assignment. If $\phi_x = \bigwedge_r \phi'_r$ then $\epsilon = 1/(2^{q+1}(q-2))$ holds. □

2.11 Polynomial-Time Algorithms for the Biclique Edge Cover and Partition Problems

This section reviews some known polynomial time algorithms that are related to the biclique edge cover and partition problems. In Subsection 2.11.1, a polynomial-time algorithm of the biclique edge cover problem for a distance-hereditary bipartite graph is noticed. In Subsection 2.11.2, a relation between the minimum biclique (cover) problem and the rank of a matrix is noticed.

2.11.1 A Polynomial-time algorithm for a distance-hereditary bipartite graph

As mentioned in Introduction, the biclique edge cover problem can be solved in polynomial time for some restricted graph classes. Müller [Mül96] showed that for C_4-free bipartite and distance-hereditary bipartite graphs, the biclique edge cover problem can be solved in polynomial time. If a given graph B is C_4-free bipartite then each maximal biclique in B is a stargraph. For this graph class, the biclique edge cover problem is equivalent to VERTEX COVER. VERTEX COVER can be solved in polynomial time when restricted to bipartite graphs [Yan81]. Thus the biclique edge cover problem can be solved in polynomial time for C_4-free bipartite graph class.

Algorithm 2 Algorithm for the distance-hereditary bipartite graph [Mül96].

```
Input a bipartite graph B = (V, E)
s ⇐ 0
F ⇐ E
Remove isolated vertices in B.
while V ≠ ∅ do
  Choose a bisimplicial edge e of B[V].
  s ⇐ s + 1
  Remove all edges dependent on e from F.
  Remove isolated vertices in B = (V, F).
end while
Output s
```

Let us overview Müller's algorithm of the biclique edge cover problem for a distance-hereditary bipartite graph [Mül96]. For a bipartite graph $B = (V, E)$, let $e = (u, v)$ be a bismplicial edge in B. That is, the induced subgraph $B[N_B(v) \cup N_B(u)]$ is a biclique of B. Müller showed that any distance-hereditary bipartite graph contains a bisimplicial edge. Let us define some terms for presenting the algorithm. A pair of edges e and f in $B = (V, E)$ is *dependent* in B if either of the following conditions holds: (1) e and f are incident with a common vertex in V, or (2) e and f form a cycle of length four in B, possibly with chords. $M \subseteq E$ is an *independent set of edges* in $B = (V, E)$ if no pair of distinct edges of M is dependent in B. The algorithm is shown in Algorithm 2. From the fact that a bisimplicial edge is contained in exactly one maximal biclique [Mül96], it is easy to show that s, the output of Algorithm 2, is the size of an optimal solution of the biclique edge cover.

The class of domino-free bipartite graphs properly contains the class of C_4-free bipartite graphs the class of the distance-hereditary bipartite graphs. Amilhastre et al. [AVJ98] showed that the size of the minimum biclique edge partition is equal to the size of the minimum biclique edge cover for this graph class. As mentioned in Chapter 1, the biclique edge cover problem can be solved in polynomial time for the domino-free bipartite graph class. Thus the biclique edge partition problem also can be solved in polynomial time for this graph class. The computational complexity of these problems for the domino-free bipartite graph class is noticed in Chapter 4.

2.11.2 The biclique edge cover and partition problems and the rank of matrix

As shown in Chapter 1, the biclique edge cover and partition problems can be formulated by using matrices. This section shows that the size of the minimum biclique edge partition is given by the rank of a 0-1 matrix, and the size of the minimum biclique edge cover is given by the Boolean rank of a matrix. It is also known that a lower bound of the minimum biclique edge partition is given by the Hermitian rank of same matrices.

In Chapter 1, we have demonstrated that the biclique edge partition and cover problems have an application in large databases that is called the association rule learning. The association rule learning is a well researched method of discovering significant relations between variables in large databases such as finding rules like that each customer who buys item A always buys B, or word X and word Y are often appeared in a web page.

Let us recall the application of databases appeared in Chapter 1. Let M be a 0-1 matrix. In Market Basket Analysis, M represents that if ith customer buys jth item then $M_{i,j} = 1$ otherwise $M_{i,j} = 0$. Under the requirement that each customer buys all items from a single shelf, we have considered the optimization problem of finding the minimum number of shelves.

Let M be as follows

$$M = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

In this case, customers $\{1, 2\}$ buy items $\{1, 2, 3, 4\}$, customers $\{3, 4\}$ buy items $\{1, 2, 3\}$ and customers $\{5, 6\}$ buy items $\{3, 4, 5, 6\}$. M can be expressed as a product of $M = S \cdot R$, where S is a 6×3 matrix and R is a 3×6 matrix, as

$$S = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

and

$$R = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

Matrix R shows that three shelves such that item $\{4\}$ on shelf 1, items $\{1, 2, 3\}$ on shelf 2 and items $\{3, 5, 6\}$ on shelf 3 satisfy the requirement. Suppose that $m \times n$ matrix M can be expressed the product of an $m \times k$ matrix and a $k \times n$ matrix. The rank of M is defined to be the minimum integer of k. In this case, it is easy to show that the minimum k is three.

Next, the 0-1 matrix M can be expressed as a bipartite graph B_M as follows. Make a vertex x_i for each ith row and a vertex y_j for each jth column of M. Put an edge (x_i, y_j) if and only if $M_{ij} = 1$. The bipartite graph B_M for M is shown in Fig.2.7. There are three bicliques B_1, B_2 and B_3 in B_M, where

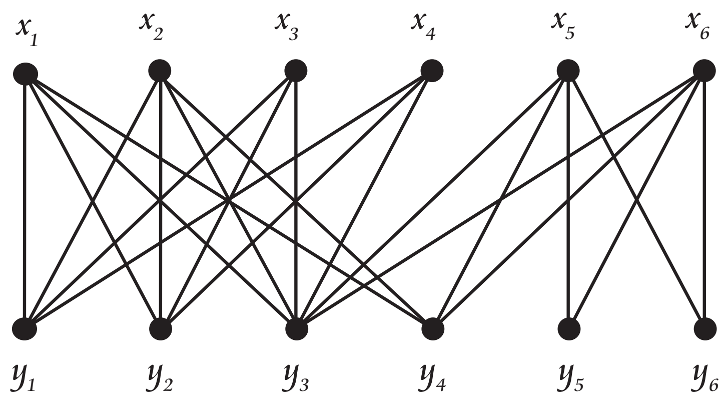

Figure 2.7: The bipartite graph B_M represents M.

$$\begin{aligned}
B_1 &= (\{x_1, x_2, x_5, x_6\}, \{y_4\}), \\
B_2 &= (\{x_1, x_2, x_3, x_4\}, \{y_1, y_2, y_3\}), \\
B_3 &= (\{x_5, x_6\}, \{y_3, y_5, y_6\}).
\end{aligned}$$

It is easy to see that $\{B_1, B_2, B_3\}$ is a biclique edge partition of B. Thus the rank of M is equal to the minimum size of the biclique edge partition of B_M.

Let us see another case of M. Let M' be a matrix as follows.

$$M' = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

M' can be expressed $S \cdot R$ where

$$S \equiv \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

and

$$R \equiv \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

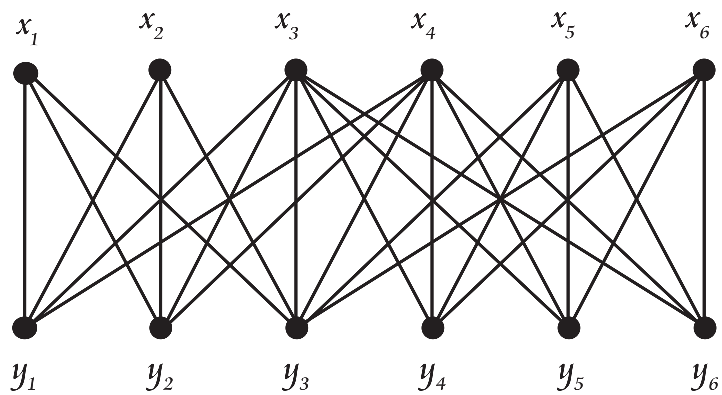

Figure 2.8: The bipartite graph $B_{M'}$ represents M'.

Same as in the case of M, the bipartite graph $B_{M'}$ is partitioned into three bi-cliques.

$$\begin{aligned} B'_1 &= (\{x_1, x_2, x_3, x_4\}, \{y_1, y_2\}), \\ B'_2 &= (\{x_1, x_2, x_3, x_4, x_5, x_6\}, \{y_3\}), \\ B'_3 &= (\{x_3, x_4, x_5, x_6\}, \{y_4, y_5, y_6\}). \end{aligned}$$

It is easy to show that this is a minimum biclique edge partition of $B_{M'}$.

Consider the following two matrices S and R.

$$S \equiv \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix},$$

$$R \equiv \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Let $\odot$ be the operator of matrix multiplication in which the addition is replaced by the Boolean addition, that is, $1 + 1 = 1 + 0 = 0 + 1 = 1$ and $0 + 0 = 0$. Then $M = S \odot R$ holds. For an $n \times m$ Boolean matrix M, the Boolean rank is defined to be the least integer k such that M can be expressed by the Boolean product ($\odot$) of $n \times k$ matrix S and $k \times m$ matrix R. In this case, it is easy to show that the minimum k is two. Thus the Boolean rank of M' is two. It is easy to show that the bipartite graph $B_{M'}$ is covered by two bicliques as follows,

$$\begin{aligned} B'_4 &= (\{x_1, x_2, x_3, x_4\}, \{y_1, y_2, y_3\}), \\ B'_5 &= (\{x_3, x_4, x_5, x_6\}, \{y_3, y_4, y_5, y_6\}). \end{aligned}$$

As we observed, the biclique edge cover problem is equivalent to the problem of finding the Boolean rank for the matrix corresponding to a given graph. On the other hand, the biclique edge partition problem is to find the "ordinary" matrix rank. Thus these statements are essentially the same: (1) the rank of 0-1 matrix is not always equal to the Boolean rank. (2) the solution of the biclique edge partition problem is not always equal to the solution of the minimum edge cover problem.

2.11.3 The biclique edge partition and the Hermitian rank

We next show a relation between the "ordinary" rank of the adjacency matrix of a general graph G and the minimum number of biclique partition $b(G)$. Let A be the adjacency matrix of G. We assume that such that $|V(G)| = 2n$. Let $n_+(A)$ be the number of positive eigenvalues and $n_-(A)$ be the number of negative eigenvalues of A. Let A be a $2n \times 2n$ Hermite matrix, that is, $A = A^T$. Then the Hermitian rank, $h(A)$ is the least k such that $A = XY^T + YX^T$ for some $2n \times k$ matrices X and Y. Then $h(A) = \max\{n_+(A), n_-(A)\}$ and Witsenhausen's inequality $b(G) \geq h(A)$ hold [GVB99]. This inequality gives a lower bound of the number of the minimum biclique edge partitioning.

We show examples of Witsenhausen's inequality for some bipartite graphs. Let A_B be the adjacency matrix of a bipartite graph $B = (X, Y, E)$ such that $|X| = |Y| = n$. Let R' be an $n \times n$ matrix such that if there is an edge (x_i, y_j) then $R'_{i,j} = 1$ otherwise $R'_{i,j} = 0$. Then

$$A_B = \begin{pmatrix} \mathbf{0} & R' \\ R'^T & \mathbf{0} \end{pmatrix}$$

where $\mathbf{0}$ is an $n \times n$ matrix with all zero entries and R'^T is the transpose of R'. Thus if $R' = R'^T$ then A_B is an Hermite matrix, and then $b(B) \geq h(A_B)$ holds.

Let B be a bipartite graph expressed by the following R'

$$R' = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}.$$

where $R' = R'^T$ holds. See Fig. 2.9. A_B is an Hermite matrix. Eigenvalues of A_B are $\{-3, -1, -1, -1, 1, 1, 1, 3\}$. Then $b(B) = 4$ and $b(B) \geq h(A_B)$ holds.

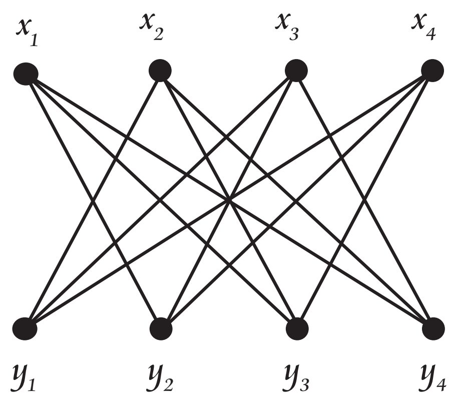

Figure 2.9: A bipartite graph expressed by B'.

A domino D (Fig.2.10) is expressed as

$$D = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}.$$

Thus the adjacency matrix A_D is a Hermite matrix. Eigenvalues of A_D are $\{-1-\sqrt{2}, -1, 1-\sqrt{2}, -1+\sqrt{2}, 1, 1+\sqrt{2}\}$. Then $h(A_D) = 3$ and $b(D) \geq h(A_D)$ holds.

As the number of eigenvalues of a matrix can be found in polynomial time, Witsenhausen's inequality gives a good estimation for a lower bound of the size of an optimal solution of the minimum bipartite edge partitioning. Gregory et.al [GVB99] showed the minimum number of bipartite edge partitioning for some graph classes using the eigenvalue of the Hermite matrix. For example, if G is the complement of a path then $b(G) = \lfloor 2(n-1)/3 \rfloor$, while if G is the complement of a cycle then $b(G) = 2\lfloor (n-1)/3 \rfloor$ or $\lfloor (2n-1)/3 \rfloor$.

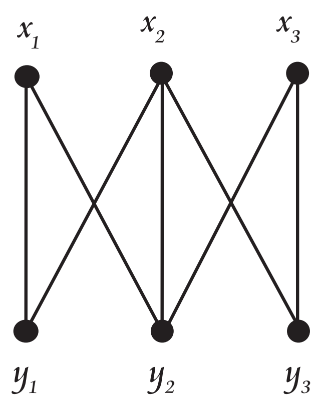

Figure 2.10: A domino D.

Chapter 3

Approximation Hardness of the Biclique Edge Partition Problem

3.1 Previous works

For a graph G, let $\mathcal{S}_{BC}(G) = \{C_1, C_2, \ldots, C_k\}$ be a collection of bicliques C_i such that each edge of G is contained in any one of C_i. The biclique edge cover problem (BEC) asks for $\mathcal{S}_{BC}(G)$ with the minimum size. (See Definition 50). Let $\mathcal{S}_{BP}(G) = \{B_1, B_2, \ldots, B_k\}$ be a collection of bicliques B_i such that each edge of G is contained in exactly one B_i. The biclique edge partition problem (BEP) asks for $\mathcal{S}_{BP}(G)$ with the minimum size. (See Definition 51). Unless P = NP, the biclique edge cover problem (BEC) does not have $O(n^{1/3})$-approximation algorithm[GH07]. It is known that BEP is NP-hard [JR93]. While BEC has been widely studied, BEP has not been given so much attention.

In this chapter, we construct a gap preserving reduction[Vaz01] from 3-OCC-MAX E2-SAT to BEP, and show for arbitrary small $\epsilon > 0$, $(6053/6052 - \epsilon)$-approximation of BEP is NP-hard. [1]

[1] Recently, the hardness result for BEP was given by Chalermsook et.al [CHHK14]. They showed that BEC and BEP do not admit $O(n^{1-\epsilon})$- and $O(m^{1/2-\epsilon})$- approximation algorithm for any $\epsilon > 0$ for a bipartite graph with n vertices and m edges, unless P = NP.

3.2 Construction of an instance of biclique edge partition problem

A Boolean expression φ is in the conjunctive normal form (CNF) if φ is a conjunction of clauses and each clause is a disjunction of literals. For a given φ in CNF, the maximum satisfiability problem (MAX SAT) asks for an assignment that satisfies simultaneously the maximum number of clauses of φ. MAX 2-SAT is MAX SAT in which each clause has at most two literals. MAX E2-SAT is MAX 2-SAT in which each clause has exactly two literals of different variables. k-OCC-MAX 2-SAT (k-OCC-MAX E2-SAT) is MAX 2-SAT (MAX E2-SAT) in which each variable occurs exactly k times in the expression.

Let N be a positive integer, Berman and Karpinski[BK98] showed inapproximability of 3-OCC-MAX 2-SAT as follows.
Theorem 5 ([BK99])**.** For any positive $\epsilon(< 1/2)$, it is NP-hard to decide whether an instance of 3-OCC-MAX 2-SAT with $2016N$ clauses has a truth assignment that satisfies at least $(2012 - \epsilon)N$ clauses, or at most $(2011 + \epsilon)N$.

In their proof, all clauses of an instance of 3-OCC-MAX 2-SAT have exactly two literals[BK98]. So this theorem can be applied to 3-OCC-MAX E2-SAT.

Let φ be an instance of 3-OCC-MAX E2-SAT and let $s(\varphi)$ be the maximum number of clauses that can be satisfied simultaneously by an assignment. In this section, we transform φ into an instance $G = (V, E)$ of BEP such that G can be partitioned into $(3 + \epsilon)m$ bicliques if and only if $s(\varphi) \geq (1 - \epsilon)m$ for a positive integer m and a positive constant ϵ (< 1).

Suppose φ has n variables $x_i(i = 1, \ldots, n)$ and m clauses $c_j(j = 1, \ldots, m)$. For $c_j = \alpha \vee \beta$, we call α (resp. β) as the first (resp. second) literal of c_j. Since each variable occurs exactly three times in φ, $3n = 2m$ holds. For each variable x_i, we construct G_i as follows.

$$\begin{aligned}
V(G_i) &= \{x_i^1, x_i^2, x_i^3, \bar{x}_i^1, \bar{x}_i^2, \bar{x}_i^3\} \\
E(G_i) &= \{(x_i^1, \bar{x}_i^2), (\bar{x}_i^2, x_i^3), \\
&\quad (x_i^3, \bar{x}_i^1), (\bar{x}_i^1, x_i^2), (x_i^2, \bar{x}_i^3), (\bar{x}_i^3, x_i^1)\}.
\end{aligned}$$

Each G_i is a cycle graph C_6 (Fig. 3.1 (a)). We denote by V_x the set of vertices in these cycles, that is, $V_x = \{x_i^d, \bar{x}_i^d | 1 \leq i \leq n, d = 1, 2, 3\}$. For each clause c_j, we create two vertices y_j, z_j and an edge $e_j = (y_j, z_j)$. Let $V_c = \{y_j, z_j | 1 \leq j \leq m\}$.

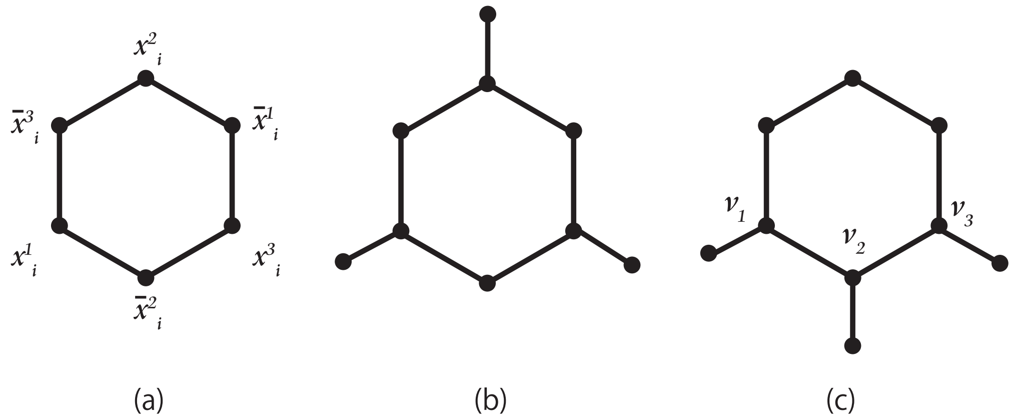

Figure 3.1: (a) G_i and its vertices. (b) G_i with nonadjacent degree-three vertices. (c) G_i with contiguous degree-three vertices.

We connect vertices of V_x and V_c as follows. Let x_i be a variable and suppose it appears in three clauses c_{j_1}, c_{j_2}, c_{j_3}. For $d = 1, 2, 3$, if the occurrence of x_i is the first literal of c_{j_d}, we connect y_{j_d} to either x_i^d (if the literal is x_i) or $\bar{x}_i^d$ (if the literal is $\bar{x}_i$) by an edge. If the occurrence of x_i is the second literal of c_{j_d}, we connect z_{j_d} to either x_i^d (if the literal is x_i) or $\bar{x}_i^d$ (if the literal is $\bar{x}_i$) by an edge. We denote by $e_{y_j}(e_{z_j})$ the added edge incident to $y_j(z_j)$.

Note that if x_i occurs all positive (all negative) in φ, the degree-three vertices appearing in G_i are not adjacent each other (Fig. 3.1 (b)). Otherwise, the degree-three vertices appear contiguously in G_i(Fig. 3.1(c)). We summarize the construction of G as follows.

$$\begin{aligned} V(G) &= V_x \cup V_c \\ E(G) &= \bigcup_{i=1}^{n} E(G_i) \cup \bigcup_{j=1}^{m} E(G_{c_j}). \end{aligned}$$

G for $\varphi = (\bar{x}_1 \vee x_2) \wedge (x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_3) \wedge (x_1 \vee x_4) \wedge (x_2 \vee x_4) \wedge (x_3 \vee \bar{x}_4)$ is shown in Fig. 3.2.

The following lemma holds.
Lemma 1. All biclique subgraphs of G are stargraphs $K_{1,s}(s \geq 1)$.

Proof. G is constructed by cycle graphs C_6 connected each other by a path P_4. Thus, G has no $K_{s,t}(s \geq 2, t \geq 2)$ as a subgraph. $\square$

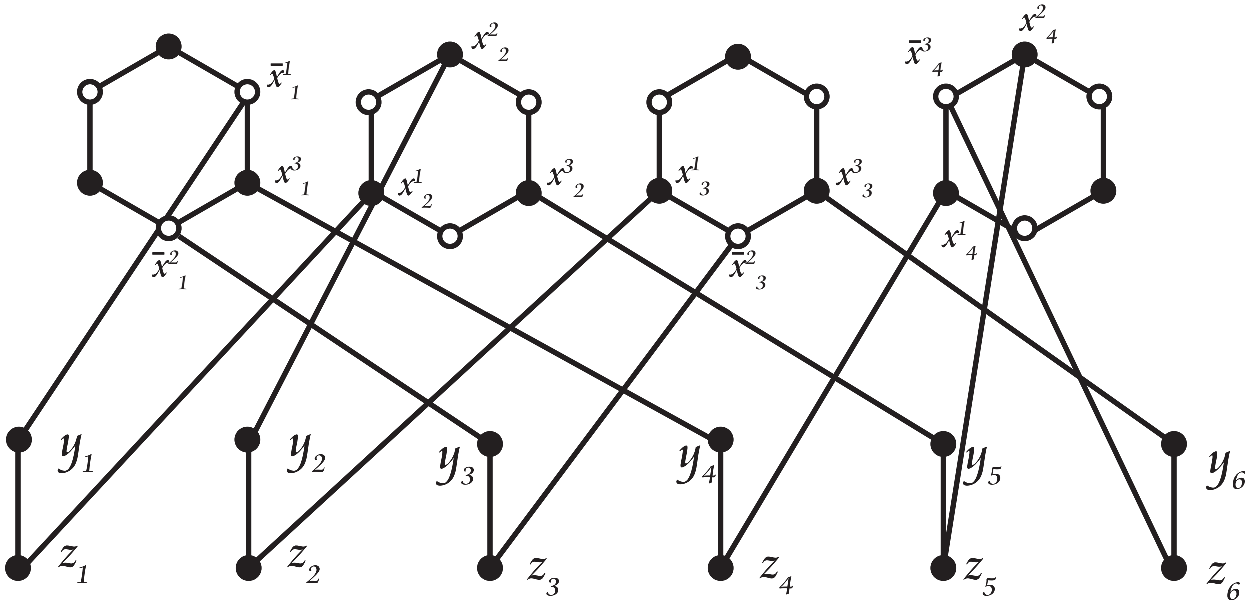

Figure 3.2: G for $\varphi = (\bar{x}_1 \vee x_2) \wedge (x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_3) \wedge (x_1 \vee x_4) \wedge (x_2 \vee x_4) \wedge (x_3 \vee \bar{x}_4)$ where black and white vertices represent positive and negative literals respectively.

3.3 Inapproximability of the biclique edge partition problem

In the sequel, we use "biclique" and "stargraph" interchangeably. For a stargraph $K_{1,s}(s \geq 2)$, the vertex of degree s is called its *center*. For each i, we define six stargraphs ($K_{1,2}$) as follows.

$$\begin{aligned}
S_i^1 &= \{(\bar{x}_i^3, x_i^1), (x_i^1, \bar{x}_i^2)\} \\
S_i^2 &= \{(\bar{x}_i^1, x_i^2), (x_i^2, \bar{x}_i^3)\} \\
S_i^3 &= \{(\bar{x}_i^2, x_i^3), (x_i^3, \bar{x}_i^1)\} \\
\bar{S}_i^1 &= \{(x_i^3, \bar{x}_i^1), (\bar{x}_i^1, x_i^2)\} \\
\bar{S}_i^2 &= \{(x_i^1, \bar{x}_i^2), (\bar{x}_i^2, x_i^3)\} \\
\bar{S}_i^3 &= \{(x_i^2, \bar{x}_i^3), (\bar{x}_i^3, x_i^1)\}.
\end{aligned}$$

We denote two sets of graphs as follows.

$$\begin{aligned}\mathcal{S}_i^T &= \{S_i^1, S_i^2, S_i^3\},\\ \mathcal{S}_i^F &= \{\bar{S}_i^1, \bar{S}_i^2, \bar{S}_i^3\}.\end{aligned}$$

Each $G_i (i = 1, \ldots, n)$ can be partitioned into the three bicliques of $\mathcal{S}_i^T$ or into the three bicliques of $\mathcal{S}_i^F$.

We denote by $S(G)$ the size of an optimal solution of BEP for G. We give the following lemma.
Lemma 2. For any positive $\epsilon(< 1)$, if $s(\varphi) > (1-\epsilon)m$ then $S(G) < (3+\epsilon)m$ holds.

Proof. Let π be an assignment that satisfies more than $(1-\epsilon)m$ clauses of φ. We show that we can construcnt a set of bicliques $\mathcal{SOL}'(G)$, a solution of BEP for G, such that $|\mathcal{SOL}'(G)| < (3+\epsilon)m$.

Let $\mathcal{SOL}'(G)$ be an empty set. For each x_i, if π assigns TRUE (FALSE) to x_i, we add $\mathcal{S}_i^T$ ($\mathcal{S}_i^F$) to $\mathcal{SOL}'(G)$. Note that all edges of G_i have been partitioned by these $3n(= 2m)$ bicliques.

Let c_j be an arbitrary clause of φ. If the assignment π satisfies c_j, there is at least one stargraph $K_{1,2}$ in $\mathcal{SOL}'(G)$ whose center is adjacent to y_j or z_j. W.l.o.g., we assume that y_j is adjacent to the center of this $K_{1,2}$. We replace this stargraph $K_{1,2}$ in $\mathcal{SOL}'(G)$ with a stargraph $K_{1,3}$ by adding e_{y_j}. This manipulation does not increase $|\mathcal{SOL}'(G)|$. Furthermore, we add to $\mathcal{SOL}'(G)$ a stargraph $K_{1,2}$ consisting of e_j and e_{z_j}.

If the assignment π does not satisfy c_j, we add two stargraphs to $\mathcal{SOL}'(G)$; $K_{1,2}$ consisting of e_j and e_{y_j}, and $K_{1,1}$ ($=e_{z_j}$). The number of $K_{1,1}$ in $\mathcal{SOL}'(G)$ is less than ϵm because of the assumption. In $\mathcal{SOL}'(G)$, we have $2m$ stargraphs, $K_{1,2}$ or $K_{1,3}$, whose centers are in V_x, and m stargraphs, $K_{1,2}$, that have an edge e_j. Thus, we have $|\mathcal{SOL}'(G)| < 2m + (1-\epsilon)m + 2\epsilon m = (3+\epsilon)m$. □

Lemma 3. For any positive $\epsilon(< 1)$, if $s(\varphi) \leq (1-\epsilon)m$ then $S(G) \geq (3+\epsilon)m$ holds.

Proof. Assume that there is a set of bicliques $\mathcal{SOL}(G)$, a solution of BEP for G, such that $|\mathcal{SOL}(G)| < (3+\epsilon)m$. We show that there is an assignment that satisfies more than $(1-\epsilon)m$ clauses of φ.

We construct $\mathcal{SOL}'(G)$, a solution of BEP for G, such that $\mathcal{SOL}'(G)$ induces an assignment that satisfies more than $(1-\epsilon)m$ clauses of φ, and $|\mathcal{SOL}'(G)| \leq |\mathcal{SOL}(G)|$

holds. Let $\mathcal{SOL}'(G)$ be an empty set. We denote by $\mathcal{SC}(G)$ the set of all bicliques in $\mathcal{SOL}(G)$ that have an edge $e_j (j = 1, \ldots, m)$. Then $|\mathcal{SC}(G)| = m$. We add all bicliques in $\mathcal{SC}(G)$ to $\mathcal{SOL}'(G)$.

Next, we remove all edges of bicliques in $\mathcal{SC}(G)$ from G. If there are singletons in the resulted graph, we remove all of them. Let G' be the resulted graph. G' consists of n connected components. Each of the connected components is G_i possibly with its incident edges. Note that for all $j(= 1 \ldots, m)$, at least one edge e_{yj} or e_{zj} remains in G'.

For each $i(= 1, \ldots, n)$, we denote by G'_i a connected component of G' whose C_6 subgraph is G_i. We denote by $\mathcal{A}$ the set of all G'_i that has no contiguous degree-three vertices, and we denote by $\mathcal{B}$ the set of all G'_i that has some contiguous degree-three vertices.

It is clear that each $G'_i \in \mathcal{A}$ cannot be partitioned into less than three bicliques. For each $G'_i \in \mathcal{A}$, we add three bicliques as shown in Fig. 3.1(a), to $\mathcal{SOL}'(G)$ as follows. If some of $x_i^d (d \in \{1, 2, 3\})$ are the degree-three vertices, we add three bicliques(stargraphs) whose centers are x_i^d to $\mathcal{SOL}'(G)$. Otherwise, we add three bicliques(stargraphs) whose centers are $\bar{x}_i^d$ to $\mathcal{SOL}'(G)$.

It is also clear that each $G'_i \in \mathcal{B}$ cannot be partitioned into less than four bicliques. For each $G'_i \in \mathcal{B}$, we add four bicliques as follows. If there are three degree-three vertices in G'_i, we denote these contiguous vertices by v_1, v_2, v_3 in this order as shown in Fig. 3.1(c). We add to $\mathcal{SOL}'(G)$ four bicliques; one stargraph $K_{1,1}$ that is an edge connecting v_2 and a vertex of e_j, two stargraphs $K_{1,3}$ whose center vertices are v_1 and v_3 and one stargraph $K_{1,2}$ for the remaining part(Fig. 3.3 (b)).

If there are only two degree-three vertices in G'_i, we denote these two contiguous vertices by v_1 and v_2. Then we add to $\mathcal{SOL}'(G)$ four bicliques; one stargraph $K_{1,1}$ that is an edge connecting v_2 and a vertex in e_j, one stargraph $K_{1,3}$ whose center vertex is v_1 and two stargraphs $K_{1,2}$ for the remaining part.

$\mathcal{SOL}'(G)$ has the same subset $\mathcal{SC}(G)$ of $\mathcal{SOL}(G)$, and the remaining part is partitioned into the optimal number of bicliques. So it is clear that $\mathcal{SOL}'(G)$ is a biclique partition of G and $|\mathcal{SOL}'(G)| \leq |\mathcal{SOL}(G)|$ holds.

Let $|\mathcal{B}| = \epsilon' m$, then $|\mathcal{A}| = n - \epsilon' m$ and

$$|\mathcal{SOL}'(G)| = |\mathcal{SC}(G)| + 3|\mathcal{A}| + 4|\mathcal{B}| = (3 + \epsilon')m.$$

From the assumption $|\mathcal{SOL}(G)| < (3 + \epsilon)m$, we have $(3 + \epsilon')m < (3 + \epsilon)m$, and $|\mathcal{B}| < \epsilon m$ holds.

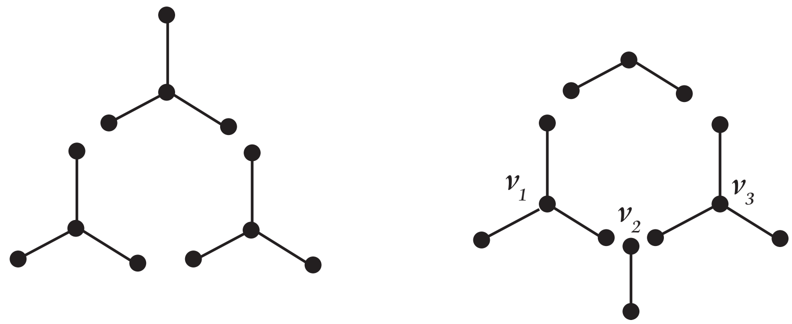

Figure 3.3: (a) G_i partitioned into three bicliques. (b) G_i partitioned into four bicliques.

We induce an assignment π' from $\mathcal{SOL}'(G)$ as follows. For each $G'_i \in \mathcal{A}$, if some of $x_i^d(\bar{x}_i^d)$ are degree-three vertices, we assign TRUE(FALSE) to x_i. If there is no degree-three vertex in G'_i, we assign FALSE to x_i. For each $G'_i \in \mathcal{B}$, if the degree-three vertex v_1 is $x_i^d(\bar{x}_i^d)$ for some $d \in \{1,2,3\}$, we assign TRUE(FALSE) to x_i.

Note that under this assignment π' the literals associating degree-three vertices denoted by v_2 are FALSE and the other literals are TRUE. Therefore, if c_j is not satisfied by π', at least one endpoint of e_j must be adjacent to v_2 in some $G'_i \in \mathcal{B}$. The number of vertices denoted by v_2 in G is exactly the size of $\mathcal{B}$. Since $|\mathcal{B}| < \epsilon m$, the number of clauses not satisfied by π' is less than ϵm, and thus π' satisfies more than $(1-\epsilon)m$ clauses in φ. $\square$

Theorem 6. $(6053/6052-\epsilon)$-approximation of BEP is NP-hard, for arbitrary small $\epsilon > 0$.

Proof. From Theorem 5, it is NP-hard to decide whether $s(\varphi) > (2016N - 4N - \epsilon N)$ or $s(\varphi) \leq (2016N - 5N + \epsilon N)$ for φ with $2016N$ clauses. Let $m = 2016N$, $\epsilon_1 m = (4+\epsilon)N$, and $\epsilon_2 m = (5-\epsilon)N$. From Lemma 2, if $s(\varphi) > (1-\epsilon_1)m$ then $S(G) < (3+\epsilon_1)m = 3 \cdot 2016N + (4+\epsilon)N$. From Lemma 3, if $s(\varphi) \leq (1-\epsilon_2)m$ then $S(G) \geq (3+\epsilon_2)m = 3 \cdot 2016N + (5-\epsilon)N$. Therefore, for any ϵ, it is NP hard to decide whether $S(G) < (6052+\epsilon)N$ or $S(G) >= (6053-\epsilon)N$. $\square$

Chapter 4

New Graph Class for the Biclique Edge Cover Problem

In this chapter, a new graph class for which the biclique edge cover problem can be solved in polynomial time is presented.

For a bipartite graph B, the modified Galois lattice $G_m(B)$ is defined. Investigating the structure of $G_m(B)$, the redundant parameter $R(B)$ is introduced. It is proved that for a bipartite graph B with $R(B) \leq 1$, the biclique edge cover problem can be solved in polynomial time.

4.1 Previous Works

The problem of covering the edges of a graph has been studied in various ways. In this chapter, we consider the cover problem in which all edges of an input bipartite graph are covered by the edges of bicliques (BEC). (See Definition 50). Covering a graph by bicliques arises in many areas [FH96]. In computer graphics, bicliques are used to model the rectangle cover problem that asks if a rectilinear polygon can be expressed as the union of a minimum number of rectangles [Lub90]. There are some applications in artificial intelligence, data mining [Wil09] and biology [NMWA78].

BEC is NP-hard for general bipartite graphs [Sto75][Orl77][FMPS09]. [1] BEC is NP-hard for chordal bipartite graphs. However, it can be solved in polynomial time for $C4$-free bipartite graphs [Mül96], bipartite distance-hereditary graphs [Mül96] and bipartite domino-free graphs [AVJ98]. A bipartite graph is $C4$-free if it has no cycle of length four as an induced subgraph. There are some characterizations for bipartite distance-hereditary graphs and we adopt the following definition: a bipartite graph is bipartite distance-hereditary if it is (6,2)-chordal, that is, every cycle of length at least 6 has at least 2 chords. A bipartite graph is domino-free if it has no domino as an induced subgraph, where a domino is a cycle of length six with exactly one chord as in Fig. 4.1. By definition, neither bipartite C_4-free graphs nor bipartite distance-hereditary graphs have any domino as an induced subgraph. Thus the class of bipartite domino-free graphs properly contains $C4$-free bipartite graphs and distance-hereditary bipartite graphs.

Amilhastre et al. [AVJ98] showed that the size of a minimum biclique cover and the size of a minimum biclique partition are equal if the graph is bipartite domino-free. To solve these problems, they defined a partial order for the set of maximal bicliques of a bipartite domino-free graph B. They used the Hasse diagram (the Galois lattice) $G(B)$ of this partial ordered set and solved the biclique edge cover/partition problem by finding a minimum cut of $G(B)$. The time complexity of this algorithm is $O(n \times m)$, where n and m are the numbers of vertices and edges of the input graph, respectively.

In this chapter, we define the modified Galois lattice $G_m(B)$ for a bipartite graph B. Next, we introduce the redundant parameter $R(B)$, and show that $R(B) = 0$ if and only if B is domino-free. Furthermore, for the input graph such that $R(B) = 1$, we show that the biclique edge cover problem can be solved in polynomial time.

In Section 4.2, we give definitions which are necessary for our discussion. Also we define the modified Galois lattice $G_m(B)$ for a bipartite graph B. In Section 4.3, some properties of $G_m(B)$ are investigated and some lemmas related to $G_m(B)$ are proved. In Section 4.4, defining the redundant parameter $R(B)$, we prove that B is a domino-free bipartite graph if and only if $R(B) = 0$. Also, we show that if $R(B) = 1$, the biclique edge cover problem can be solved in polynomial time.

[1] Chalermsook et.al [CHHK14] showed that BEC do not admit $O(n^{1-\epsilon})$- and $O(m^{1/2-\epsilon})$-approximation algorithm for any $\epsilon > 0$ for a bipartite graph with n vertices and m edges, unless $\mathrm{P} = \mathrm{NP}$.

4.2 Definitions

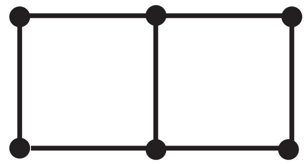

Figure 4.1: A domino

Let $B = (X, Y, E)$ be a bipartite graph, where $X = \{x_1, x_2, \ldots, x_{n_x}\}$, $Y = \{y_1, y_2, \ldots, y_{n_y}\}$ are the sets of vertices and $E \subseteq X \times Y$ is the set of edges. Let $n = n_x + n_y$ and $m = |E_B|$. Let $N_B(x) = \{y \mid (x, y) \in E_B\}$ be the set of neighbors of x in B. We denote by X_B the set X of vertices of B. We denote by Y_B the set Y of vertices of B. Let $K = (X', Y', E')$ be a subgraph of B such that $X' \subseteq X_B$ and $Y' \subseteq Y_B$. K is a *biclique* if $E' = X' \times Y'$, $X' \neq \emptyset$ and $Y' \neq \emptyset$. A *biclique edge cover* of B is a set of bicliques $\{K_1, K_2, \ldots, K_s\}$ such that $E_B = \bigcup_{i=1}^{s} E_{K_i}$, and a *biclique edge partition* of B is a set of bicliques $\{K_1, K_2, \ldots, K_s\}$ such that $E_B = \bigcup_{i=1}^{s} E_{K_i}$ and $E_{K_i} \cap E_{K_j} = \emptyset\,(i \neq j)$. In this chapter, we simply call the biclique edge cover (partition) as "biclique cover (partition)".

A *domino* is a cycle of length six with exactly one chord that produces two $C4$'s as in Fig. 4.1. A bipartite graph B is *domino-free* if B has no domino as an induced subgraph. Let $\mathcal{K}_M(B)$ be the set of maximal bicliques of B. We define a partially order $<$ on $\mathcal{K}_M(B)$ as follows. For distinct bicliques $K_p, K_q \in \mathcal{K}_M(B)$, $K_p < K_q$ if and only if $Y_{K_p} \subset Y_{K_q}$ (See Fig. 4.2). K_r and K_s are *incomparable* if neither $K_r < K_s$ nor $K_s < K_r$. Let $(\mathcal{K}_M(B), \leq)$ be the reflexive closure of the defined ordered set.

In [AVJ98], Amilhastre et al. defined a directed graph $G(B)$ for a domino-free bipartite graph B as follows. The set of vertices of $G(B)$ is $\mathcal{K}_M(B) \cup \{\top, \bot\}$, where $\top$ is the maximum element to $\mathcal{K}_M(B)$, that is, $\top > K$ for all $K \in \mathcal{K}_M(B)$ and $\bot$ is the minimum element. For two elements K_p and K_q such that $K_p < K_q$, put a directed edge (K_q, K_p) if there is no K_r such that $K_p < K_r$ and $K_r < K_q$. They call $G(B)$ as Galois lattice of B [AVJ98]. $G(B)$ is actually the Hasse diagram of the partially ordered set $(\mathcal{K}_M(B), \leq)$ [Wil92].

In this section, we define the modified Galois lattice $G_m(B)$ as follows. Here, we do not assume that B is domino-free. Let $X_i\,(1 \leq i \leq n_x)$ be the maximal stargraph centered at x_i. Note that X_i may not be a maximal biclique in B. Denote the set

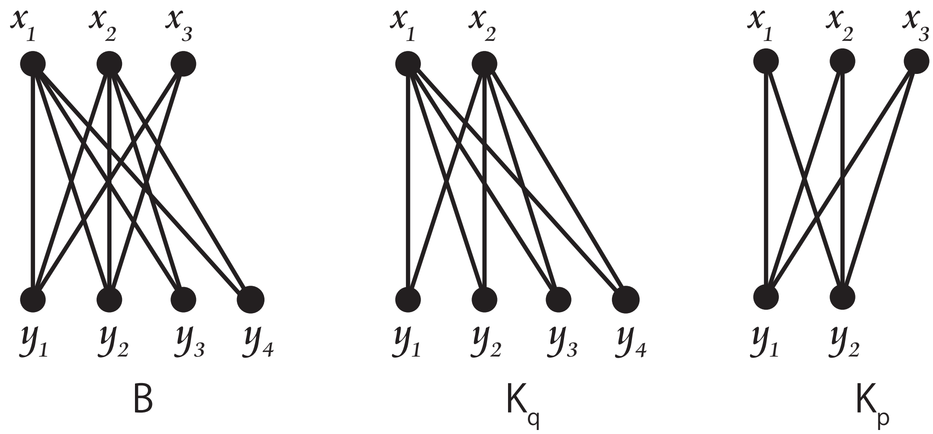

Figure 4.2: Two bicliques K_p and K_q in B such that $K_p < K_q$.

of all X_i by $\mathcal{X}_s(B)$. Define $Y_j\,(1 \leq j \leq n_y)$ and $\mathcal{Y}_s(B)$ in the same manner. We define the partial order on $\mathcal{K}_s(B) \equiv \mathcal{K}_M(B) \cup \mathcal{X}_s(B) \cup \mathcal{Y}_s(B)$ as follows: for any distinct $K_p, K_q \in \mathcal{K}_s(B)$, $K_p < K_q$ if and only if $Y_{K_p} \subseteq Y_{K_q}$ and $X_{K_p} \supseteq X_{K_q}$. Let $\mathcal{K}(B) \equiv \mathcal{K}_s(B) \cup \{\top, \bot\}$. According to this partial order on $\mathcal{K}(B)$, we construct $G_m(B)$ in the same manner as $G(B)$. In the rest of this chapter, $<$ represents the partial order defined in this paragraph.

Here we give some examples of the modified Galois lattice. Fig. 4.3 and Fig. 4.4 show that a bipartite graph B and its modified Galois lattice. ($\top$ and $\bot$ are omitted.) Here, we follow the conventional drawing of the Hasse diagram, that is, each edge has downward direction. It is easy to see that the graph B in Fig. 4.3 has two maximal bicliques and the graph B in Fig. 4.4 has three maximal bicliques.

Fig. 4.5 shows the case in which B is itself a domino and its modified Galois lattice.

Let us see an example for a bipartite graph B shown in Fig. 4.6. As vertices $\{x_2, x_3, x_4, y_3, y_4, y_5\}$ induces a domino, B is not domino-free. It is obvious that

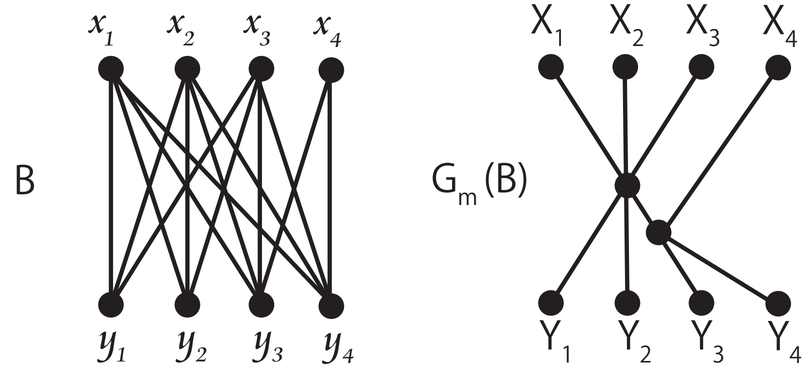

Figure 4.3: B with two maximal bicliques and its modified Galois lattice $G_m(B)$.

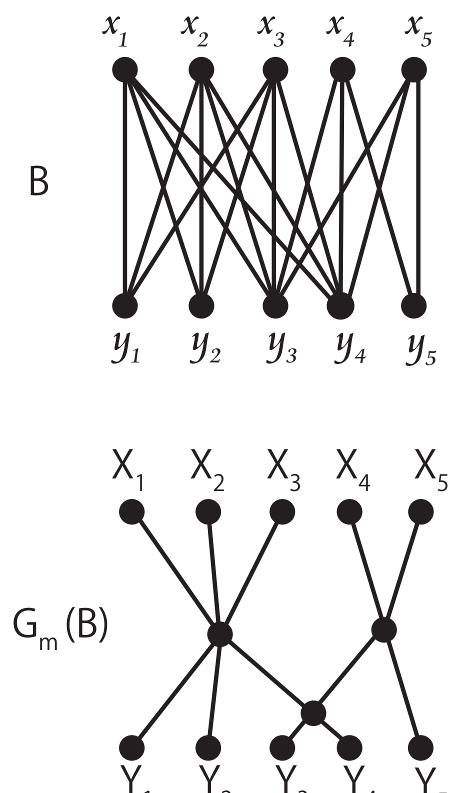

Figure 4.4: B with three maximal bicliques and its modified Galois lattice $G_m(B)$.

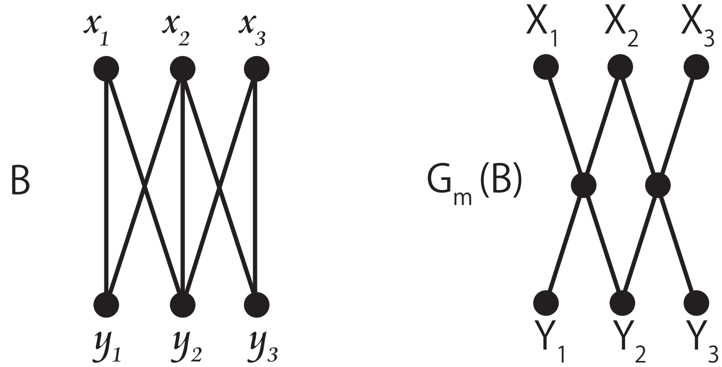

Figure 4.5: A domino B and its modified Galois lattice $G_m(B)$.

B has six maximal bicliques $K_1, \ldots, K_6$ such that

$$\begin{aligned}
X_{K_1} &= \{x_1, x_4\},\ Y_{K_1} = \{y_1, y_2, y_3\},\\
X_{K_2} &= \{x_2, x_3\},\ Y_{K_2} = \{y_2, y_3, y_4\},\\
X_{K_3} &= \{x_3, x_4\},\ Y_{K_3} = \{y_2, y_3, y_5, y_6\},\\
X_{K_4} &= \{x_1, x_2, x_3, x_4\}, Y_{K_4} = \{y_2, y_3\},\\
X_{K_5} &= \{x_3\},\ Y_{K_5} = \{y_2, y_3, y_4, y_5, y_6\}\\
X_{K_6} &= \{x_4\}, \text{ and } Y_{K_6} = \{y_1, y_2, y_3, y_5, y_6\}.
\end{aligned}$$

Then the Galois lattice $G(B)$ and the modified Galois lattice $G_m(B)$ are shown in Fig. 4.7 and Fig. 4.8, respectively.

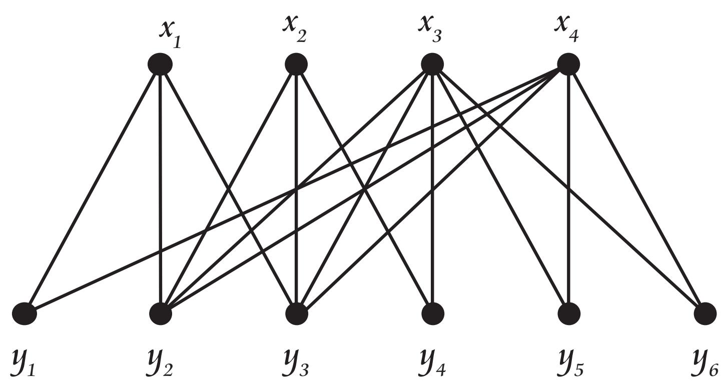

Figure 4.6: A Bipartite graph B.

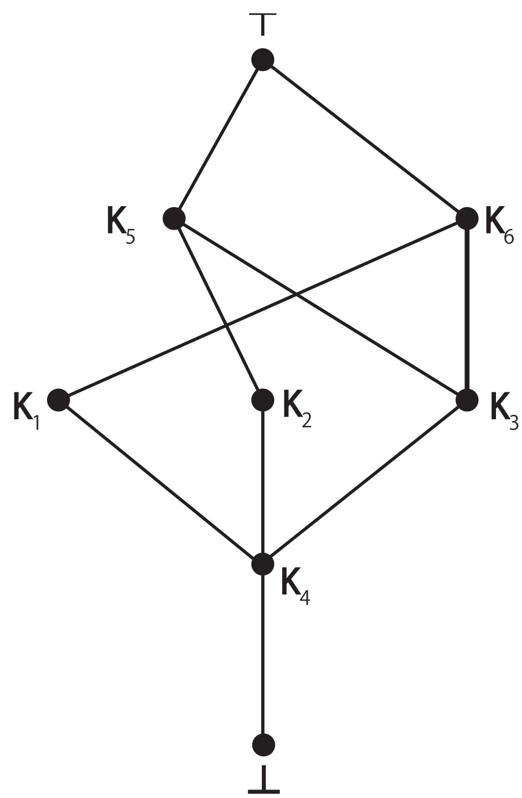

Figure 4.7: The Galois lattice $G(B)$ for B in Fig. 4.6.

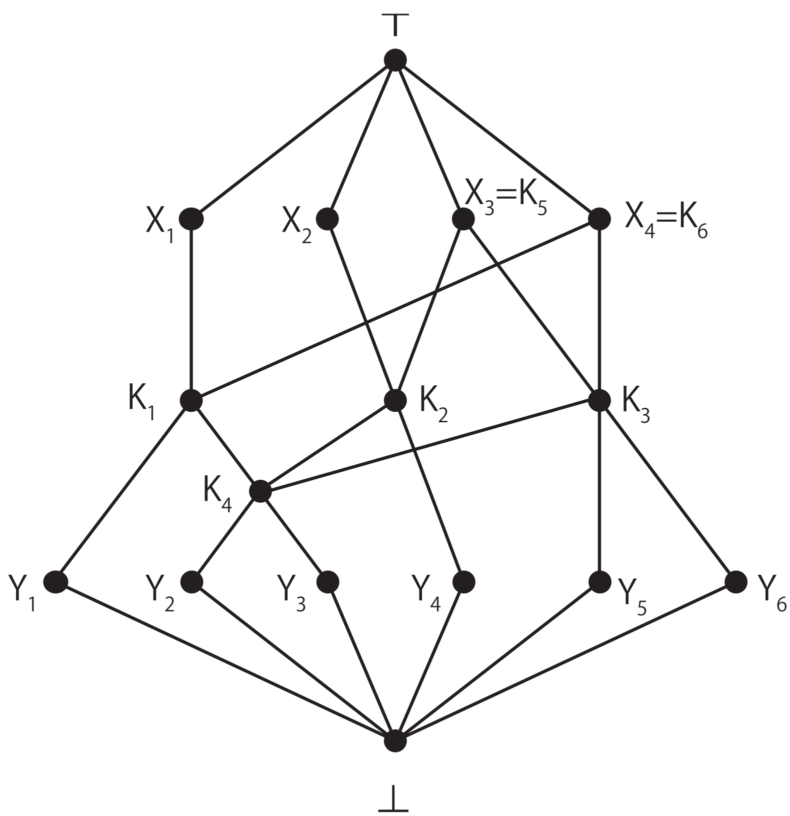

Figure 4.8: The modified Galois lattice $G_m(B)$ for B in Fig. 4.6.

Note that the Galois lattice is embedded, in some way, in the modified Galois lattice.

Amilhastre et al. [AVJ98] defined a "simplification" operation on a domino-free bipartite graph. They repeatedly apply this operation to an input bipartite graph B until no operation can be applied. The resulted graph is called as a "simplified" domino-free bipartite graph. $G_m(B)$ is coincident with $G(B)$ if B is a simplified domino-free bipartite graph.

4.3 Properties of the modified Galois lattice

Let $K_1 = (X_{K_1}, Y_{K_1}, E_{K_1})$ and $K_2 = (X_{K_2}, Y_{K_2}, E_{K_2})$ be different bicliques in $\mathcal{K}_M(B)$. K_1 and K_2 have the following property.
Property 1. For any distinct $K_1, K_2 \in \mathcal{K}_M(B)$, $X_{K_1} \subset X_{K_2} \iff Y_{K_2} \subset Y_{K_1}$.

Proof. Assume that $X_{K_1} \subset X_{K_2}$ and $Y_{K_2} \not\subset Y_{K_1}$. Then there exists a vertex $y \in Y_{K_2}$ but $y \notin Y_{K_1}$. From $y \in Y_{K_2}$, $(x, y) \in E_{K_2} \subseteq E_B$ follows for any $x \in X_{K_2}$. From $X_{K_1} \subset X_{K_2}$, biclique $K_3 = (X_{K_1}, Y_{K_1} \cup \{y\}, X_{K_1} \times (Y_{K_1} \cup \{y\}))$ is a subgraph of B, which contradicts to the maximality of K_1. □

For two vertices $X_i \in \mathcal{X}_s(B)$ and $Y_j \in \mathcal{Y}_s(B)$ of $G_m(B)$, let $\mathcal{P}(i, j)$ be the set of directed paths from X_i to Y_j. (Note that X_i is a stargraph in B and it is a vertex in $G_m(B)$.) Then we have the next lemma.
Lemma 4. $|\mathcal{P}(i, j)| > 0 \iff (x_i, y_j) \in E_B$, for all i and j.

Proof. ($\Rightarrow$) Assume that there is a directed edge from X_i to Y_j in $G_m(B)$. Then $\{y_j\} = Y_{Y_j} \subseteq Y_{X_i} = N_B(x_i)$ holds. Thus there is edge (x_i, y_j) in B. Assume that there is a path $P \in \mathcal{P}(i, j)$ from X_i to Y_j with length greater than two. Let $P = (X_i, K_{i_1}, \ldots, K_{i_s}, Y_j)$. Then $X_i > K_{i_1} > \ldots > K_{i_s} > Y_j$ and thus $X_i > Y_j$ holds. This means that in B, the center of stargraph Y_j is in $Y_{X_i}(= N_B(x_i))$. Therefore, B has edge (x_i, y_j).

($\Leftarrow$) Assume that B has an edge (x_i, y_j). Then $y_j \in N_B(x_i)$, and thus, $Y_{Y_j} \subseteq Y_{X_i}$ and $Y_j < X_i$. Therefore, there is at least one directed path from X_i to Y_j in $G_m(B)$.

□

We have the following lemmas for a vertex on a path from a vertex of $\mathcal{X}_s(B)$ to a vertex of $\mathcal{Y}_s(B)$ in $G_m(B)$.
Lemma 5. Let K be a vertex on a path from X_i to Y_j then $(x_i, y_j) \in E_K$.

Proof. If K is either X_i or Y_j then the lemma obviously holds. Then K is not a stargraph and $X_i > K > Y_j$ holds. Therefore, in B, $Y_{X_i} \supseteq Y_K \supset Y_{Y_j} = \{y_j\}$ holds. Thus, (x_i, y_j) is an edge of K, since K is a maximal biclique. □

Lemma 6. If $(x_i, y_j) \in E_K$ for some $K \in \mathcal{K}(B)\backslash\{\top, \bot\}$ then there is a path from X_i to Y_j passing through K in $G_m(B)$.

Proof. Since $(x_i, y_j) \in E_K$, $x_i \in X_K$. Then $X_{X_i} \subseteq X_K$ and $Y_K \subseteq Y_{X_i}$. Thus $K \leq X_i$ holds. Similarly, $Y_j \leq K$ holds. From the construction of $G_m(B)$, there is a path from X_i to K and a path from K to Y_j. □

Let $\mathcal{C}$ be a subset of $\mathcal{K}(B)\backslash\{\top, \bot\}$. $\mathcal{C}$ is a *cut* of $G_m(B)$, if for all i, j, every path from X_i to Y_j on $G_m(B)$ has at least one vertex that belongs to $\mathcal{C}$. That is, all paths from a vertex of $\mathcal{X}_s(B)$ to a vertex of $\mathcal{Y}_s(B)$ are cut by $\mathcal{C}$. Obviously $\{X_1, \ldots, X_{n_x}\}$ (or also $\{Y_1, \ldots, Y_{n_y}\}$) is a cut of $G_m(B)$. A minimum cut of $G_m(B)$ is a cut with the minimum size. In Fig. 4.8, for example, $\{K_1, K_2, K_3\}$ is the minimum cut of $G_m(B)$.
Lemma 7. A cut of $G_m(B)$ is a biclique cover of B.

Proof. Let $\mathcal{C}$ be a cut of $G_m(B)$. For any $(x_i, y_j) \in E_B$, there is a path from vertex $X_i \in \mathcal{X}_s(B)$ to vertex $Y_j \in \mathcal{Y}_s(B)$ in $G_m(B)$ by Lemma 4. Let K be a vertex on the path and $K \in \mathcal{C}$. From Lemma 5, K has edge (x_i, y_j). Thus, every edge (x_i, y_j) of B is contained in at least one biclique of C.

□

If B is a domino-free bipartite graph, then B has the following property. (We give the proof to make the chapter self-contained.)
Property 2 (Theorem 3.1 of [AVJ98])**.** Let B be a bipartite graph. Then B is domino-free if and only if for any distinct $K_1, K_2 \in \mathcal{K}_M(B)$ such that K_1 and K_2 have at least one common edge, one of these statements is true: (i) $X_{K_1} \subset X_{K_2}$ and $Y_{K_2} \subset Y_{K_1}$,(ii) $X_{K_2} \subset X_{K_1}$ and $Y_{K_1} \subset Y_{K_2}$.

Proof. ($\Rightarrow$) Let K_1 and K_2 be two maximal bicliques sharing a common edge $\{x, y\}$ and such that (i) and (ii) are false. From Property 1, we have $X_{K_1}\backslash X_{K_2} \neq \emptyset$, $X_{K_2}\backslash X_{K_1} \neq \emptyset$, $Y_{K_1}\backslash Y_{K_2} \neq \emptyset$ and $Y_{K_2}\backslash Y_{K_1} \neq \emptyset$. Let $x_1 \in X_{K_1}\backslash X_{K_2}$. We claim that there exists $y_2 \in Y_{K_2} \backslash Y_{K_1}$ such that $(x_1, y_2) \notin E_B$. If $Y_{K_2} \backslash Y_{K_1} \subseteq N(x_1)$ then $Y_{K_2} \subseteq N(x_1)$ since $Y_{K_1} \cap Y_{K_2} \subseteq N(x_1)$. Then K_2 is not maximal. Thus, there exists $y_2 \in Y_{K_2} \backslash Y_{K_1}$ such that $(x_1, y_2) \notin E_B$. Let $x_2 \in X_{K_2} \backslash X_{K_1}$. From similar discussion, there exist $y_1 \in Y_{K_1} \backslash Y_{K_2}$ such that $(x_2, y_1) \notin E_B$. Then, $\{x, y, x_1, y_1\}$ and $\{x, y, x_2, y_2\}$ induces two C_4's. As $(x_1, y_2) \notin E_B$ and $(x_2, y_1) \notin E_B$ holds, $\{x, y, x_1, y_1, x_2, y_2\}$ induces a domino.

($\Leftarrow$) Assume that B has a domino induced by $\{x, y, x_1, y_1, x_2, y_2\}$ with chord $\{x, y\}$. Then there is $K_1 \in \mathcal{K}_M(B)$ such that K_1 contains $C_4 = (x, y, x_1, y_1)$ and $K_2 \in \mathcal{K}_M(B)$ such that K_2 contains $C_4 = (x, y, x_2, y_2)$. Since $(x_1, y_2) \notin E_B$, $x_1 \in X_1 \backslash X_2$, so (i) is false. Similarly, we obtain that (ii) is false.

□

We define *Unique Path Condition* as follows.

$$\begin{gathered}\text{For all } i, j \, (1 \leq i \leq n_x, 1 \leq j \leq n_y) \\ |\mathcal{P}(i, j)| = 1 \iff (x_i, y_j) \in E_B.\end{gathered}$$

Lemma 8. If B is a domino-free bipartite graph then $G_m(B)$ satisfies Unique Path Condition.

Proof. Suppose B is a domino-free bipartite graph. From Lemma 4, $(x_i, y_i) \in E_B$, if and only if $|\mathcal{P}(i, j)| \geq 1$. Therefore it is sufficient to prove that if $(x_1, y_i) \in E_B$ then $|\mathcal{P}(i, j)| < 2$.

Assume that $|\mathcal{P}(i, j)| \geq 2$. Let P_1, P_2 be paths from X_i to Y_j such that $P_1 \neq P_2$. Then there are two incomparable bicliques K_1 on P_1 and K_2 on P_2. Note that neither K_1 nor K_2 is a stargraph. Thus $|X_{K_1}|, |Y_{K_1}|, |X_{K_2}|, |Y_{K_2}| \geq 2$ holds. Since K_1 and K_2 are incomparable, neither $Y_{K_1} \subset Y_{K_2}$ nor $Y_{K_2} \subset Y_{K_1}$. Thus $Y_{K_1} \backslash Y_{K_2} \neq \emptyset$ and $Y_{K_2} \backslash Y_{K_1} \neq \emptyset$ hold. As K_1 and K_2 are maximal bicliques, Property 1 implies that $X_{K_1} \backslash X_{K_2} \neq \emptyset$ and $X_{K_2} \backslash X_{K_1} \neq \emptyset$. Then there exist four vertices of B, x_1, x_2, y_1 and y_2 such that $x_1 \in X_{K_1}$, $x_1 \notin X_{K_2}$, $x_2 \notin X_{K_2}$, $x_2 \in X_{K_1}$, $y_1 \in Y_{K_1}$, $y_1 \notin Y_{K_2}$, $y_2 \notin Y_{K_1}$ and $y_2 \in Y_{K_2}$. Thus, the graph induced by the set of vertices $\{x_i, x_1, x_2, y_j, y_1, y_2\}$ is a domino. This contradicts to the premise that B is a domino-free bipartite graph. Therefore, if $(x_i, y_j) \in E_B$ then $|\mathcal{P}(i, j)| = 1$.

□

Also the converse of Lemma 8 holds.
Lemma 9. If Unique Path Condition holds then B is a domino-free bipartite graph.

Proof. Assume that B is not a domino-free graph. Then there is a subgraph induced by six vertices of two C_4's sharing edge (x_i, y_j). Then there are two incomparable maximal bicliques K_1 and K_2 that shares edge (x_i, y_j). Thus there is two distinct paths form X_i to Y_j in $G_m(B)$ and $|\mathcal{P}(i,j)| \geq 2$ holds. That is, if B is not a domino-free graph, then Unique Path Condition does not hold.

$\square$

Let $\mathcal{P}$ be the set of all paths from a vertex of $\mathcal{X}_s(B)$ to a vertex of $\mathcal{Y}_s(B)$ in $G_m(B)$, that is, $\mathcal{P} = \bigcup_{1\leq i\leq n_x, 1\leq j\leq n_y} \mathcal{P}(i,j)$. Let $P_{i,j} \in \mathcal{P}(i,j)$ be a path from X_i to Y_j. Let f be a map from $\mathcal{P}$ to E_B such that $f(P_{i,j}) \to (x_i, y_j)$. For example, in Fig. 4.8, a path $P = (X_2, K_2, K_4, Y_3)$ is mapped to edge (x_2, y_3), that is, $f(P) = (x_2, y_3)$.
Corollary 7. *B is a domino-free bipartite graph if and only if f is bijective.*

Proof. From Lemma 8 and Lemma 9, the corollary holds.

$\square$

For any biliques K_1, K_2 in B, we define a subgraph $K_{2-1} = K_2 - K_1$. K_{2-1} has all edges of K_2 but no edge of K_1, and has no singletons. We denote the edges of K_{2-1} by E_{2-1}. From Property 2, the next lemma holds.
Lemma 10. (Lemma 3.1 of [AVJ98]) *Let B be a domino-free bipartite graph. Let K_1 be any maximal biclique and K_2 be any biclique in B such that $E_{K_2} \not\subset E_{K_1}$. Then K_{2-1} is a biclique.*

Proof. If K_2 is a stargraph, the proof is trivial. Assume that K_2 is not a stargraph. Let $K_3 \in \mathcal{K}_M(B)$ such that $E_{K_2} \subseteq E_{K_3}$. By Property 2, there are two cases: (i) $X_{K_3} \subset X_{K_1}$ and (ii)$Y_{K_3} \subset Y_{K_1}$. (i) $X_{K_3} \subset X_{K_1}$ implies $X_{K_2} \subset X_{K_1}$. Then for any $x \in X_{K_2}$ and $y \in Y_{K_2} \backslash Y_{K_1}$, $(x, y) \in E_{2-1}$ and $(x, y) \notin E_{K_1}$ holds. Thus $K_{2-1} = (X_{K_2}, Y_{K_2} \backslash Y_{K_1}, E_{2-1})$ is a biclique of B. (ii) $Y_{K_3} \subset Y_{K_1}$ implies $Y_{K_2} \subset Y_{K_1}$. Then for any $x \in X_{K_2} \backslash X_{K_1}$ and $y \in Y_{K_1}$, $(x, y) \in E_{2-1}$ and $(x, y) \notin E_{K_1}$ holds. Thus $K_{2-1} = (X_{K_2} \backslash X_{K_1}, Y_{K_2}, E_{2-1})$ is a biclique of B. There is no other case.

$\square$

Theorem 8. (Theorem 3.2 of [AVJ98]) *Let B be a domino-free bipartite graph. The size of a minimum biclique cover of B is equal to the size of a minimum biclique partition of B.*

Proof. Let $\mathcal{S}_{\text{COVER}}(B)$ be a minimum biclique cover of B and let $\mathcal{S}_{\text{PARTITION}}(B)$ be a minimum biclique partition of B. Since any biclique partition of B is also a biclique cover of B, $|\mathcal{S}_{\text{COVER}}(B)| \leq |\mathcal{S}_{\text{PARTITION}}(B)|$ holds. Let $\mathcal{S}_{\text{COVER}}(B) = \{K_1, K_2, \ldots, K_c\}$. Then $\{K_i - K_{i+1} - K_{i+2} - \cdots - K_c | 1 \leq i \leq c\}$ is a set of bicliques of B (Lemma 10) that form a biclique partition of B. Thus $|\mathcal{S}_{\text{COVER}}(B)| \geq |\mathcal{S}_{\text{PARTITION}}(B)|$ holds. Therefore $|\mathcal{S}_{\text{COVER}}(B)| = |\mathcal{S}_{\text{PARTITION}}(B)|$. □

A *cut* of $G_m(B)$ is a set of vertices $\{k_1, k_2, \ldots, k_t\}$ of $G_m(B)$ such that there is at least one k_i $(1 \leq i \leq t)$ of every path from $\top$ to $\bot$. Let $\mathcal{S}_{\text{CUT}}(B)$ be a cut of $G_m(B)$ with the size is the minimum. Then the next theorem holds.

Theorem 9. Let B be a domino-free bipartite graph. Then $|\mathcal{S}_{\text{CUT}}(B)| = |\mathcal{S}_{\text{PARTITION}}(B)| = |\mathcal{S}_{\text{COVER}}(B)|$ holds.

Proof. From Theorem 8, it is sufficient to prove that $|\mathcal{S}_{\text{CUT}}(B)| = |\mathcal{S}_{\text{COVER}}(B)|$. Assume that there is a path from X_i to Y_j in $G_m(B)$. Then there exists an edge (x_i, y_j) in B by Lemma 4. As $\mathcal{S}_{\text{COVER}}(B)$ covers (x_i, y_j), there exists $K \in \mathcal{S}_{\text{COVER}}(B)$ such that $(x_i, y_j) \in E_K$. From Lemma 6, K is on a path from X_i to Y_j in $G_m(B)$. If B is a domino-free bipartite graph, then the path from X_i to Y_j in $G_m(B)$ is unique from Corollary 7. Thus, $\mathcal{S}_{\text{COVER}}(B)$ is a cut of $G_m(B)$ and $|\mathcal{S}_{\text{CUT}}(B)| \leq |\mathcal{S}_{\text{COVER}}(B)|$. From Lemma 7, $|\mathcal{S}_{\text{CUT}}(B)| \geq |\mathcal{S}_{\text{COVER}}(B)|$ holds. Therefore, $|\mathcal{S}_{\text{CUT}}(B)| = |\mathcal{S}_{\text{COVER}}(B)|$.

□

For a simplified domino-free bipartite graph B, Amilhastre et al. [AVJ98] showed that the size of Galois lattice $G(B)$ is $O(n + m)$. They constructed $G(B)$ in $O(n \times m)$ time. Since a minimum cut of $G(B)$ can be computed in polynomial time by using network flows techniques, the minimum cover/partition problem can be solved in polynomial time.

4.4 The redundant parameter and the biclique edge cover

We denote the degree of a vertex x in B by $d_B(x)$. We denote by $\mathcal{P}(i, *)$ the set of directed paths of $G_m(B)$ from X_i to any vertex of $\mathcal{Y}_s(B)$, and denote by $\mathcal{P}(*, j)$ the set of directed paths of $G_m(B)$ from any vertex of $\mathcal{X}_s(B)$ to Y_j. That is, $\mathcal{P}(i, *) = \cup_{j=1}^{n_y} \mathcal{P}(i, j)$ and $\mathcal{P}(*, j) = \cup_{i=1}^{n_x} \mathcal{P}(i, j)$.

We define $R(B)$ as follows.

$$R(B) \equiv \max\big(\max_{1\le i\le n_x}(|\mathcal{P}(i,*)| - d_B(x_i)), \max_{1\le j\le n_y}(|\mathcal{P}(*,j)| - d_B(y_j))\big)$$

We call it the *redundant parameter* of B. For example, for B in Fig. 4.6, it is easy to verify that $R(B) = 2$.

Theorem 10. B is a domino-free bipartite graph if and only if $R(B) = 0$.

Proof. Assume that B is a domino-free bipartite graph. From Corollary 7, there is a bijective map such that the unique path from X_i to Y_j is mapped to edge (x_i, y_j). Thus $|\mathcal{P}(i,*)|$ is the number of the edges incident to x_i and $|\mathcal{P}(i,*)| = d_B(x_i)$ holds for all i. Similarly, $|\mathcal{P}(*,j)| = d_B(y_j)$ holds for all j. Therefore, $R(B) = 0$ holds.

Assume that $R(B) = 0$. As $|\mathcal{P}(i,*)| \ge d_B(x_i)$, $R(B) = 0$ implies $|\mathcal{P}(i,*)| = d_B(x_i)$ for all i. From Lemma 4, there is an unique path in $\mathcal{P}(i,*)$ from X_i to each Y_j such that $(x_i, y_j) \in E_B$. Then f is a bijective map from $\mathcal{P}$ to E_B. Therefore B is a domino-free bipartite graph by Corollary 7. □

If $R(B) = 0$ then B is a domino-free bipartite graph, and any minimum cut of $G_m(B)$ defines a minimum cover/partition of B by Theorem 9. We will show that if $R(B) = 1$, any minimum cover of B is a minimum cut of $G_m(B)$. Note that the minimum cover of B does not define the minimum partition of B, if B is not domino-free. For example, B in Fig. 4.5 can be covered by two bicliques, but cannot partitioned into less than three bicliques.

Theorem 11. Let B be a bipartite graph with $R(B) \le 1$. Then any biclique cover of B is a cut of $G_m(B)$.

Proof. Assume that there is a minimum biclique cover $\mathcal{S}$ of B that is not a cut of $G_m(B)$. As $\mathcal{S}$ is not a cut, there is at least one path P that is not cut by $\mathcal{S}$ in $G_m(B)$. Let P be a path from X_1 to Y_1 in $G_m(B)$. Since edge (x_1, y_1) is covered by $\mathcal{S}$, if there is no vertex on P except for X_1 and Y_1, then X_1 or Y_1 is in $\mathcal{S}$. This contradicts to the assumption that P is not cut by $\mathcal{S}$. Thus there is at least one biclique K on P. Since $\mathcal{S}$ does not cut P, $K \notin \mathcal{S}$. As K is not a stargraph, it has at least four vertices that induce C_4 in B. Let $x_1, x_2 \in X_K$ and $y_1, y_2 \in Y_K$ and $e_1 = (x_1, y_1)$, $e_2 = (x_1, y_2)$, $e_3 = (x_2, y_1)$ and $e_4 = (x_2, y_2)$. As $\mathcal{S}$ is a cover of B, these four edges must be covered by some bicliques K_i in $\mathcal{S}$. There are two cases that we must consider.

(Case 1) Assume that $\mathcal{S}$ has four distinct bicliques $K_1, \ldots, K_4$ such that $e_i \in E_{K_i}$ and $e_i \notin E_{K_{i'}}$ for $i \neq i'$. Then there are eight vertices such that

$$\begin{aligned}
x_1, x_3 &\in X_{K_1},\, y_1, y_3 \in Y_{K_1}, \\
x_1, x_4 &\in X_{K_2},\, y_2, y_4 \in Y_{K_2}, \\
x_2, x_5 &\in X_{K_3},\, y_1, y_5 \in Y_{K_3}, \\
x_2, x_6 &\in X_{K_4},\, y_2, y_6 \in Y_{K_4}.
\end{aligned}$$

See Fig. 4.9 and Fig. 4.10. Since $e_1 = (x_1, y_1) \in E_{K_1}$, K_1 is on a path P' from X_1 to Y_1 from Lemma 6. $K_1 \in \mathcal{S}$ implies $P' \neq P$. Thus the number of paths from X_1 to Y_1 is at least two. Similar discussion holds for K_2, thus the number of paths from X_1 to Y_2 is at least two. Therefore the number of paths from X_1 to Y_1 or Y_2 is at least four. From Lemma 4, there is a path from x_1 to each $y_j \in N_B(x_1)$. Thus $R(B) \geq |\mathcal{P}(1, *)| - d_B(x_1) \geq 2$ holds.

(Case 2) Assume that there is a biclique $K_1 \in \mathcal{S}$ such that K_1 has at least two edges among $e_i\,(i = 1 \ldots 4)$. Without loss of generality, we can assume that K_1 has e_1, e_2. (See Fig. 4.11 and Fig. 4.12.) Since $e_1 = (x_1, y_1) \in E_{K_1}$, K_1 is on a path P' from X_1 to Y_1 by Lemma 6 and $K \neq K_1$. Thus the number of paths from X_1 to Y_1 is at least two. Since $e_2 = (x_1, y_2) \in E_K$, K is on a path P_1 from X_1 to Y_2. Since $e_2 = (x_1, y_2) \in E_{K_1}$, K_1 is on a path $P_1{}'$ from X_1 to Y_2. Thus the number of paths from X_1 to Y_2 is at least two. Therefore, there are at least four paths from X_1 to Y_1 or Y_2. From Lemma 4, there is a path from x_1 to each $y_j \in N_B(x_1)$. Thus $R(B) \geq |\mathcal{P}(1, *)| - d_B(x_1) \geq 2$ holds.

Therefore, if $R(B) \leq 1$ the assumption that $\mathcal{S}$ is not a cut of $G_m(B)$ fails.

$\square$

Theorem 11 is the best one in the sense that there is a bipartite graph B with $R(B) = 2$ for which the theorem does not hold. For example, the graph shown in Fig. 4.9 can be covered by $\{K_1, K_2, K_3, K_4\}$, but this set is not a cut of $G_m(B)$ (Fig. 4.10). Whereas for the graph B shown in Fig. 4.13, it is easy to verify that $R(B) = 1$ and a cut of B is a cover of B.

Corollary 12. Let B be a bipartite graph with $R(B) \leq 1$. Then any minimum cut of $G_m(B)$ is a minimum biclique cover of B.

Proof. Let C be a minimum cut of $G_m(B)$. From Lemma 7, C is a biclique cover of B. Let $\mathcal{S}_{\text{COVER}}(B)$ be a minimum biclique cover of B. Then $|\mathcal{S}_{\text{COVER}}(B)| \leq |C|$.

From Theorem 11, $\mathcal{S}_{\mathrm{COVER}}(B)$ is a cut of $G_m(B)$. This implies $|\mathcal{S}_{\mathrm{COVER}}(B)| \geq |C|$. Therefore, $|\mathcal{S}_{\mathrm{COVER}}(B)| = |C|$, and thus C is a minimum biclique cover of B.

$\square$

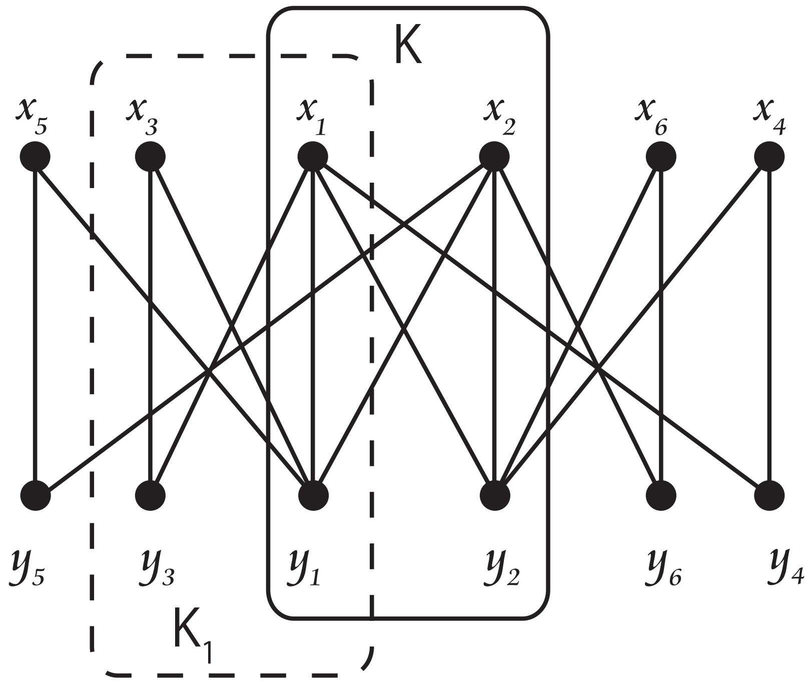

Figure 4.9: K and K_1 in Case 1.

In the rest of this chapter, we investigate the size of $G_m(B)$. $G_m(B)$ could be very large if B is not domino-free. Consider the bipartite graph $B = K_{n,n} - M_n$, where $K_{n,n}$ is the complete bipartite graph with $2n$ vertices and M_n is its perfect matching. Then B has $2^n - 2$ maximal bicliques, and thus $G_m(B)$ has 2^n vertices. If $R(B) = 0$, that is, B is domino-free, then the number of edges in $G(B)$ is $O(n + m)$ [AVJ98] and also it is $O(n + m)$ in $G_m(B)$.

We will show that for a bipartite graph B with $R(B) = 1$ the number of edges in

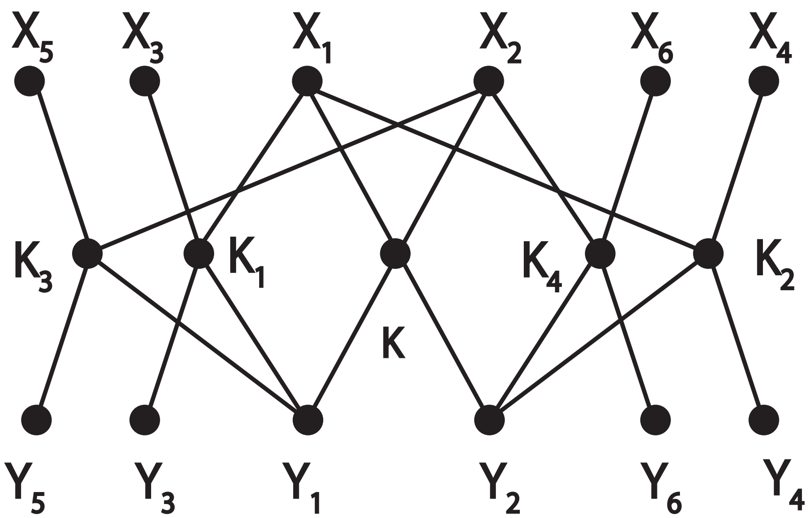

Figure 4.10: The modified Galois lattice of the graph in Fig. 4.9 (excluding $\top$ and $\perp$).

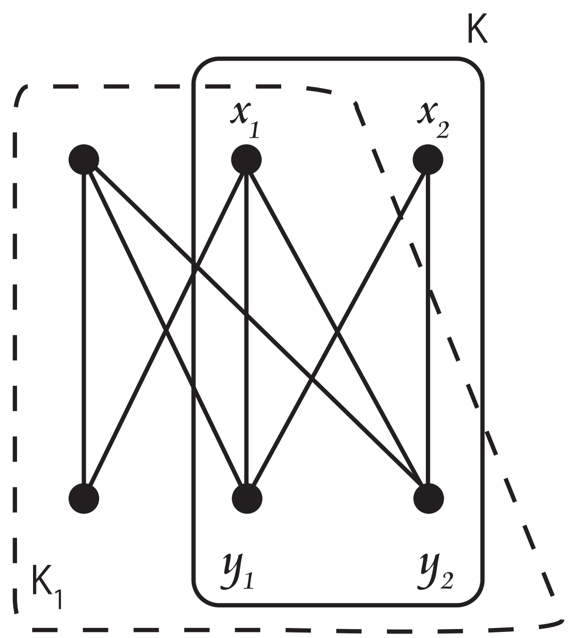

Figure 4.11: The subgraphs of K and K_1 induced by x_1, x_2, y_1, y_2 in Case 2.

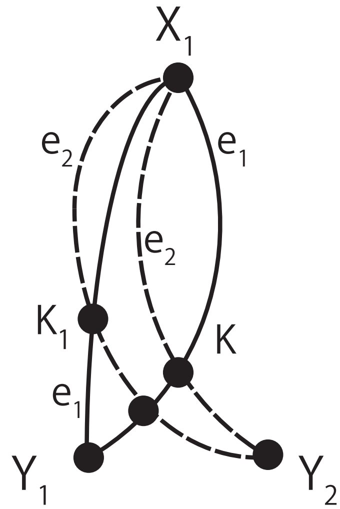

Figure 4.12: Two paths from X_1 to Y_1 in $G_m(B)$ for Fig. 4.11.

$G_m(B)$ is bounded by $2n+m$. Assume $R(B)=1$, we have

$$\begin{aligned}\sum_{i=1}^{n_x} |\mathcal{P}(i,*)| &\leq \sum_{i=1}^{n_x}(R(B)+d_B(x_i)) = n_x + m,\\ \sum_{j=1}^{n_y} |\mathcal{P}(*,j)| &\leq \sum_{j=1}^{n_y}(R(B)+d_B(y_j)) = n_y + m.\end{aligned}$$

Thus, the total number of paths from vertices of $\mathcal{X}_s(B)$ to vertices of $\mathcal{Y}_s(B)$ is at most $n+m$. Then next theorem holds.

Theorem 13. Let B be a bipartite graph with $R(B)=1$. Then the number of edges in $G_m(B)$ is at most $2n+m$.

Proof. We replace all vertices in $G_m(B)$ that are not stargraphs with bicliques as follows. Let $K \in \mathcal{K}_M(B)$ be a vertex in $G_m(B)$. Let $X_K = \{x_1,\ldots,x_s\}$, $Y_K = \{y_1,\ldots,y_t\}$ in B. Delete K and its incident edges from $G_m(B)$, and add edges $X(K)\times Y(K)$ where $X(K)=\{X_1,\ldots,X_s\}$ and $Y(K)=\{Y_1,\ldots,Y_t\}$. Note that we allow multiedges when we add edges. In this operation, the number of edges does not decrease in $G_m(B)$. The number of paths from $\top$ to $\bot$ does not change and is bounded by $2n+m$. Thus, after replacing all vertices of $\mathcal{K}_M(B)$, the total number of the edges in $G_m(B)$ is equal to the total number of the paths. Note that if we replace

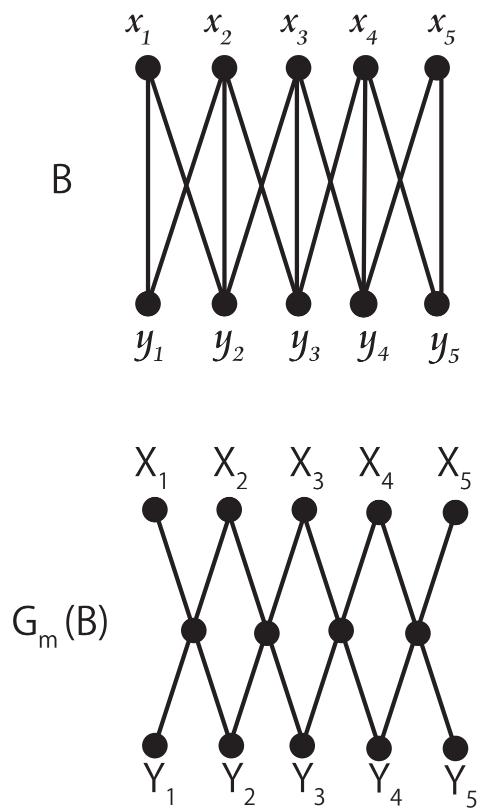

Figure 4.13: A graph B with $R(B) = 1$ and the modified Galois lattice $G_m(B)$.

each multiedge with a single edge and delete $\top$ and $\perp$ and their incident edges, we obtain B. Therefore, the lemma holds.

$\square$

4.5 Summary

In this chapter, we define the modified Galois lattice $G_m(B)$ for a bipartite graph B. We introduce the redundant parameter $R(B)$, and show that $R(B) = 0$ if and only if B is a domino-free. Furthermore, we show that the biclique edge cover problem can be solved in polynomial time for the class of bipartite graphs B with $R(B) = 1$. This graph class properly includes the domino-free bipartite graphs.

Chapter 5

The Size of the Modified Galois Lattice $G_m(B)$

As we have seen in Section 4.4, the size of the modified Galois lattice can be very large for a general bipartite graph. Here, we review the example given in Section 4.4. Consider a bipartite graph $B_n = (X, Y, E)$ with $|X| = |Y| = n$ vertices. Let $X = \{x_1, x_2, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$. B_n has edges (x_i, y_j) for all i and j such that $i \neq j$. Let I be any subset of $\{1, 2, \ldots, n\}$. Let $X' = \bigcup_{i \in I} x_i$ and $Y' = \bigcup_{j \in \{1,2,\ldots,n\}-I} y_j$. That is, X' is a subset of X and Y' is the subset of Y uniquely determined by X'. Then the subgraph induced by $X' \cup Y'$ is a maximal biclique. Thus the size (the number of vertices) of $G_m(B_n)$ increases exponentially to n.

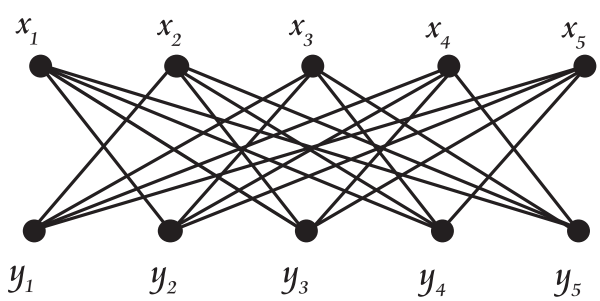

Figure 5.1: The bipartite graph B_5.

In this chapter, it is shown that the size of the modified Galois lattice is fairly small for some restricted graph classes. For a distance-hereditary bipartite graph B with n vertices, it is shown that the size of $G_m(B)$ is at most $2n + 1$. Furthermore, for general bipartite graphs with n vertices and m edges, it is shown that the size of the modified Galois lattice is at most $(R(B)/2 + 1)n + m + 2$.

A distance-hereditary bipartite graph contains a bisimplicial edge [Mül96]. As mentioned in Section 2.11.1, every bisimplicial edge is contained in only one maximal biclique. The following lemma is given in [Mül96] and the proof is given for the reader's convenience.

Lemma 11. Let B be a distance-hereditary bipartite graph. Let e be a bisimplicial edge in $K_i \in \mathcal{K}_M(B)$. Then $e \notin E(K_j)$ for $K_j \in \mathcal{K}_M(B)$ $(j \neq i)$.

Proof. Let $e = (x, y)$ be a bisimplicial edge. Assume that $e \in K_1$ and $e \in K_2$ for $K_1, K_2 \in \mathcal{K}_M(B)$ and $K_1 \neq K_2$. As e is a bisimplicial edge, the subgraph induced by $N(x) \cup N(y)$ is a biclique. From our assumption, there are four distinct vertices x_1, y_1, x_2, y_2 such that $x_1 \in X_{K_1} - X_{K_2}$, $y_1 \in Y_{K_1} - Y_{K_2}$, $x_2 \in X_{K_2} - X_{K_1}$, and $y_2 \in Y_{K_2} - Y_{K_1}$. Then vertices x, y, x_1, y_1, x_2, y_2 induce C_6 with a cord. This contradicts that B is a distance hereditary bipartite graph, that is, $(6, 2)$-chordal graph. □

Recall that edge e and e' are dependent if either e and e' share an endpoint in common or there is a cycle C_4 that has e and e'. An edge e and e' are independent if they are not dependent. We say that an edge e and a biclique K are independent if any edge in K and e are independent. If $e = (x, y)$ and K are independent then $x \notin X_K$ and $y \notin Y_K$ hold. Let $B = (X, Y, E)$ be a distance-hereditary bipartite graph with $n(= |X| + |Y| \geq 2)$ vertices. Let $\mathcal{K}_M(B)$ be the set of all maximal bicliques of B. For any $v \in B$, we denote by S_v the maximal stargraph with its center v. Then the next lemma holds.

Lemma 12. For a distance-hereditary bipartite graph B with $n \geq 2$ vertices, $|\mathcal{K}_M(B)| \leq n - 1$ holds.

Proof. The proof is by induction on n. Suppose that $n = 2$ and $|E(B)| = 1$. Then the lemma holds since $|\mathcal{K}_M(B)| = 1$. If B consists of two isolated vertices, then $|\mathcal{K}_M(B)| = 0$, and thus the lemma holds.

Let B be a distance-hereditary bipartite graph with $n > 2$ vertices. We can assume that B is connected. Let $e = (x, y)$ be a bisimplicial edge in B. From Lemma 11, e is contained in the only one maximal biclique K. We claim that at least one of x and y is in only one maximal biclique. Assume that there are two different maximal bicliques K_1 and K_2 and $x \in V(K_1)$ and $y \in V(K_2)$. Then $e \in E(S_x)$ and $e \in E(S_y)$, and both of S_x and S_y are maximal bicliques of B. This leads to a contradiction since e is a bisimplicial edge. Thus at least one of x and y is in only one maximal biclique.

Let x be contained in only K. Delete x from B and let B' be the resulted graph. Then B' is a distance-hereditary graph from the definition of distance-hereditary graphs.

A maximal biclique in B that is independent to e, is also a maximal biclique in B'. A maximal biclique in B that is not K and dependent to e, is also a maximal biclique in B'. The subgraph induced by $V(K) \setminus \{x\}$ may not be a maximal biclique in B'. Thus the decrease of the number of vertices in B' is at most one. That is $|\mathcal{K}_M(B')| \geq |\mathcal{K}_M(B)| - 1$. From the assumption of the induction, $|\mathcal{K}_M(B')| \geq n-2$ holds for B'. Therefore, $n - 1 \geq |\mathcal{K}_M(B')| + 1 \geq |\mathcal{K}_M(B)|$. □

Corollary 14. For a distance-hereditary bipartite graph B with $n(\geq 2)$ vertices, $|V(G_m(B))| \leq 2n+1$ holds.

Proof. Since $V(G_m(B)) = V(\mathcal{K}_M(B)) \cup V(\mathcal{X}_s(B)) \cup V(\mathcal{Y}_s(B))$, $|V(G_m(B))| \leq |\mathcal{K}_M(B)| + n + 2 \leq 2n+1$ holds. □

For a general bipartite graph, we have the next theorem.

Theorem 15. $|V(G_m(B))| \leq (R(B)/2+1)n + m + 2$ holds for any bipartite graph B.

Proof. Let $\mathcal{M}(B) \equiv \mathcal{K}_M(B) \backslash \{\mathcal{X}_s(B) \cup \mathcal{Y}_s(B)\}$. We claim that the number of paths from a vertex in $\mathcal{X}_s(B)$ to a vertex in $\mathcal{Y}_s(B)$ is greater than the size of $\mathcal{M}(B)$. To prove this claim, we consider a mapping from $\mathcal{M}(B)$ to a set of paths of $G_m(B)$.

We denote by $\mathcal{P}$ the set of all paths from a vertex in $\mathcal{X}_s(B)$ to a vertex in $\mathcal{Y}_s(B)$ of $G_m(B)$. That is, $\mathcal{P} = \cup_{j=1}^{n_y} \mathcal{P}(*, j)$. Let $P \in \mathcal{P}$ be a path of $G_m(B)$ from a vertex X in $\mathcal{X}_s(B)$ to a vertex Y in $\mathcal{Y}_s(B)$. Let $K_0(=X), K_1, K_2, \ldots, K_t, K_{t+1}(=Y)$ be the vertices on P in $G_m(B)$ from X to Y in this order. Then $X_{K_1} \subset X_{K_2} \subset \cdots \subset X_{K_t}$ holds from the maximality of bicliques. For each $j\,(1 \leq j \leq t)$, choose a vertex $x_{i_j} \in V(B)$ such that $x_{i_j} \in X_{K_j} \setminus X_{K_{j-1}}$. Since $x_{i_j} \in X_{K_j}$, there is a path $P' \in \mathcal{P}$ from X_{i_j} to K_j in $G_m(B)$. Let P_{K_j} be the path obtained by concatinating P' and the subpath of P from K_j to Y in $G_m(B)$. We map each K_j to P_{K_j} for $1 \leq j \leq t$. For each remaining paths in $\mathcal{P}$, we map a vertex K_i on the path to $P_{K_i} \in \mathcal{P}$ in the same way unless P_{K_i} is already mapped from some K_j in the previous construction of mapping. From the construction of this mapping, it is clear that each vertex in $\mathcal{M}(B)$ is mapped to a distinct path of $\mathcal{P}$. Thus for each vertex $K \in \mathcal{M}(B)$, there is at least one distinct path from a vertex in $\mathcal{X}_s(B)$ to a vertex in $\mathcal{Y}_s(B)$. From the definition of $R(B)$,

$$
\begin{aligned}
|\mathcal{M}(B)| - m \leq \sum_{i=1}^{|X|}(|\mathcal{P}(i,*)| - d_B(x_i)) &\leq R(B)|X|, \text{ and} \\
|\mathcal{M}(B)| - m \leq \sum_{i=1}^{|Y|}(|\mathcal{P}(*,j)| - d_B(y_j)) &\leq R(B)|Y|
\end{aligned}
$$

hold. Thus $2|\mathcal{M}(B)| \leq R(B)n + 2m$ and $|V(G_m(B))| \leq (R(B)/2 + 1)n + m + 2$ holds. $\square$

Then the next corollary holds.
Corollary 16. For a bipartite graph B with $R(B) = 1$, $|V(G_m(B))| = 3n/2+m+2$ holds

Gély et al. [GNS09] gave an algorithm that outputs all maximal bicliques of an input graph $G = (U, V, E)$ in lexicographical order on U with $O((|U| + |V|)^2)$ delay. If $R(B) = 1$, as the number of vertices of $G_m(B)$ is $O(n + m)$, then $G_m(B)$ can be constructed in $O(n^3 + m^3)$ time. By using network flow techniques [AMO93], the minimum cut of $G_m(B)$ can be computed in $O(|E|\sqrt{|V|})$ for a graph $G_m(B) = (V, E)$. Thus the minimum cut of $G_m(B)$ can be found in $O(n^{5/2} + m^{5/2})$ time.

Chapter 6

Conclusion

In this thesis, two closely related problems, the biclique edge partition problem and the biclique edge cover problem, have studied.

In chapter 3, it has shown that the biclique edge partition problem cannot be approximated within a ratio of 6053/6052 unless P = NP. In order to obtain this approximation hardness, an approximation-preserving (gap-preserving) reduction from 3-OCC-MAX 2-SAT is presented. Our reduction implies that if there is a 6053/6052-approximation algorithm for the biclique edge partition problem then P = NP. Therefore, under the assumption that P $\neq$ NP, the biclique edge partition problem does not have a 6053/6052-approximation algorithm. It follows that the biclique edge partition problem has no polynomial-time approximation scheme unless P = NP. To the author's knowledge, the ratio of 6053/6052 is the first explicit lower bound for the approximation hardness of the biclique edge partition problem. [1]

In chapter 4, the complexity of the biclique edge cover problem has studied. For general bipartite graphs, the biclique edge cover problem cannot be solved in polynomial time if P $\neq$ NP. This thesis has presented a new graph class for which the minimum biclique edge cover problem can be solved in polynomial time, and has shown that this graph class properly contains the domino-free bipartite graph class. In order to present the new graph class, the modified Galois lattice $G_m(B)$ for an input bipartite graph B is introduced. A partial order on the set of maximal bicliques in B is defined and $G_m(B)$ is the Hasse diagram of this partial order. Furthermore, the redundant parameter $R(B)$ is defined on $G_m(B)$. It is shown that $R(B) = 0$ if and only if B

[1] However, after this work was published, much better result was reported by Chalermsook et al. [CHHK14].

is domino-free bipartite. Furthermore, it is shown that there is a polynomial time algorithm for a graph B with $R(B) = 1$. If $R(B) > 0$ then B has at least one domino as an induced subgraph. Thus, we have a new graph class such that there is a polynomial time algorithm for the minimum biclique edge cover problem.

The computation time of the proposed algorithm depends on the size of $G_m(B)$. This thesis shows that $G_m(B)$ has at most $2n + 1$ vertices for a distance-hereditary bipartite graph B. For a graph B such that $R(B) \leq 1$, it is shown that $G_m(B)$ has at most $O(n + m)$ vertices.

Whether the biclique edge cover problem can be solved in polynomial time for graphs with $R(B) > 1$ is an open question.

Bibliography

[AMO93] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows: Theory, algorithms, and applications*. Prentice Hall, 1st edition, February 1993.

[AS98] S. Arora and S. Safra. Probabilistic checking of proofs: a new characterization of NP. *J. ACM*, 45(1):70–122, January 1998.

[AVJ98] J. Amilhastre, M. C. Vilarem, and P. Janssen. Complexity of minimum biclique cover and minimum biclique decomposition for bipartite domino-free graphs. *Discrete Appl. Math.*, 86(2-3):125–144, 1998.

[BK98] P. Berman and M. Karpinski. On some tighter inapproximability results. *ECCC TR-98-029*, 1998.

[BK99] P. Berman and M. Karpinski. On some tighter inapproximability results (extended abstract). *Lecture Notes In Computer Science*, 1644:200 – 209, 1999.

[CHHK14] P. Chalermsook, S. Heydrich, E. Holm, and A. Karrenbauer. Nearly tight approximability results for minimum biclique cover and partition. In *Algorithms - ESA 2014*, volume 8737 of *Lecture Notes in Computer Science*, pages 235–246. 2014.

[Coo71] A. S. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, pages 151–158, New York, 1971. ACM.

[Din07] I. Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3), June 2007.

[DS05] I. Dinur and S. Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, 162:439–485, 2005.

[Fei98] U. Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45(4):634–652, July 1998.

[FGL+96] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Interactive proofs and the hardness of approximating cliques. *J. ACM*, 43(2):268–292, March 1996.

[FH96] P. C. Fishburn and P. L. Hammer. Bipartite dimensions and bipartite degrees of graphs. *Discrete Mathematics*, 160(1-3):127 – 148, 1996.

[FMPS09] H. Fleischner, E. Mujuni, D. Paulusma, and S. Szeider. Covering graphs with few complete bipartite subgraphs. *Theoretical Computer Science*, 410(21 - 23):2045 – 2053, 2009.

[GH07] H. Gruber and M. Holzer. Inapproximability of nondeterministic state and transition complexity assuming P $\neq$ NP. *Lecture Notes In Computer Science*, 4588:205–216, 2007.

[GNS09] A. Gély, L. Nourine, and B. Sadi. Enumeration aspects of maximal cliques and bicliques. *Discrete Applied Mathematics*, 157(7):1447 – 1459, 2009.

[GVB99] D. A. Gregory, L. W. Valerie, and L. S. Bryan. Biclique decompositions and hermitian rank. *Linear Algebra and its Applications*, 292(1-3):267–280, 1999.

[Hås99] J. Håstad. Clique is hard to approximate within $n^{(1-\epsilon)}$. *Acta Mathematica*, 182(1):105–142, 1999.

[Hås01] J. Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, July 2001.

[JR93] T. Jiang and B. Ravikumar. Minimal NFA problems are hard. *SIAM Journal on Computing*, 22(6):1117–1141, 1993.

[KZ97] H. Karloff and U. Zwick. A 7/8-approximation algorithm for max 3sat? In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, FOCS '97, pages 406–415, Washington, DC, USA, 1997.

[Lub90] A. Lubiw. The boolean basis problem and how to cover some polygons by rectangles. *SIAM J. Discret. Math.*, 3(1):98–115, January 1990.

[Mül96] H. Müller. On edge perfectness and classes of bipartite graphs. *Discrete Mathematics*, 149(1 - 3):159 – 187, 1996.

[NMWA78] D. S. Nau, G. Markowsky, M. A. Woodbury, and D. B. Amos. A mathematical analysis of human leukocyte antigen serology. *Mathematical Biosciences*, 40(3-4):243 – 270, 1978.

[Orl77] J. Orlin. Contentment in graph theory: covering graphs with clique. *Indagationes Mathematicae (Proceedings)*, 80(5):406 – 424, 1977.

[Sto75] L. Stockmeyer. The set basis problem is NP-complete. Technical Report Technical Report RC-5431, IBM, 1975.

[Vaz01] V. V. Vazirani. *Approximation algorithms*. Springer-Verlag, 2001.

[Wil92] R. Wille. Concept lattices and conceptual knowledge systems. *Computers and Mathematics with Applications*, 23(6 - 9):493 – 515, 1992.

[Wil09] R. Wille. Restructuring lattice theory: an approach based on hierarchies of concepts. In *Formal concept analysis*, volume 5548 of *Lecture Notes in Computer Science*, pages 314–339. 2009.

[Yan81] M. Yannakakis. Node-deletion problems on bipartite graphs. *SIAM Journal on Computing*, 10(2):310–327, 1981.