Recognizing generating subgraphs revisited

Vadim E. Levit
Department of Computer Science
Ariel University, ISRAEL
levitv@ariel.ac.il

David Tankus
Department of Software Engineering
Sami Shamoon College of Engineering, ISRAEL
davidt@sce.ac.il

Abstract

A graph $G$ is well-covered if all its maximal independent sets are of the same cardinality. Assume that a weight function $w$ is defined on its vertices. Then $G$ is $w$-well-covered if all maximal independent sets are of the same weight. For every graph $G$, the set of weight functions $w$ such that $G$ is $w$-well-covered is a vector space, denoted as $W CW(G)$. Deciding whether an input graph $G$ is well-covered is co-NP-complete. Therefore, finding $W CW(G)$ is co-NP-hard.

A generating subgraph of a graph $G$ is an induced complete bipartite subgraph $B$ of $G$ on vertex sets of bipartition $B X$ and $B Y$, such that each of $S \cup B X$ and $S \cup B Y$ is a maximal independent set of $G$, for some independent set $S$. If $B$ is generating, then $w(B X) = w(B Y)$ for every weight function $w \in W CW(G)$. Therefore, generating subgraphs play an important role in finding $W CW(G)$.

The decision problem whether a subgraph of an input graph is generating is known to be NP-complete. In this article we prove NP-completeness of the problem for graphs without cycles of length 3 and 5, and for bipartite graphs with girth at least 6. On the other hand, we supply polynomial algorithms for recognizing generating subgraphs and finding $W CW(G)$, when the input graph is bipartite without cycles of length 6. We also present a polynomial algorithm which finds $W CW(G)$ when $G$ does not contain cycles of lengths 3, 4, 5, and 7.

1 Introduction

1.1 Basic Definitions and Notation

Throughout this paper $G$ is a simple (i.e., a finite, undirected, loopless and without multiple edges) graph with vertex set $V(G)$ and edge set $E(G)$.
Cycles of $k$ vertices are denoted by $C_k$. When we say that $G$ does not contain $C_k$ for some $k \geq 3$, we mean that $G$ does not admit subgraphs isomorphic to $C_k$. It is important to mention that these subgraphs are not necessarily induced.

Let $u$ and $v$ be two vertices in $G$. The distance between $u$ and $v$, denoted as $d(u, v)$, is the length of a shortest path between $u$ and $v$, where the length of a path is the number of its edges. If $S$ is a non-empty set of vertices, then the distance between $u$ and $S$, denoted as $d(u, S)$, is defined by

$$d(u, S) = \min \{d(u, s) : s \in S\}.$$  

For every positive integer $i$, denote

$$N_i(S) = \{x \in V(G) : d(x, S) = i\},$$  

and

$$N_i[S] = \{x \in V(G) : d(x, S) \leq i\}.$$  

We abbreviate $N_1(S)$ and $N_1[S]$ to be $N(S)$ and $N[S]$, respectively. If $S$ contains a single vertex, $v$, then we abbreviate $N_i(\{v\})$, $N_i[\{v\}]$, $N(\{v\})$, and $N[\{v\}]$ to be $N_i(v)$, $N_i[v]$, $N(v)$, and $N[v]$, respectively. We denote by $G[S]$ the subgraph of $G$ induced by $S$. For every two sets, $S$ and $T$, of vertices of $G$, we say that $S$ dominates $T$ if $T \subseteq N[S]$.

### 1.2 Well-Covered Graphs

Let $G$ be a graph. A set of vertices $S$ is independent if its elements are pairwise nonadjacent. An independent set of vertices is maximal if it is not a subset of another independent set. An independent set of vertices is maximum if the graph does not contain an independent set of a higher cardinality.

The graph $G$ is well-covered if every maximal independent set is maximum [14]. Assume that a weight function $w : V(G) \rightarrow \mathbb{R}$ is defined on the vertices of $G$. For every set $S \subseteq V(G)$, define $w(S) = \sum_{s \in S} w(s)$. Then $G$ is $w$-well-covered if all maximal independent sets of $G$ are of the same weight.

The problem of finding a maximum independent set in an input graph is NP-hard. However, if the input is restricted to well-covered graphs, then a maximum independent set can be found polynomially using the greedy algorithm. Similarly, if a weight function $w : V(G) \rightarrow \mathbb{R}$ is defined on the vertices of $G$, and $G$ is $w$-well-covered, then finding a maximum weight independent set is a polynomial problem.

The recognition of well-covered graphs is known to be co-NP-complete. This is proved independently in [5] and [17]. In [3] it is proven that the problem remains co-NP-complete even when the input is restricted to $K_{1,4}$-free graphs. However, the problem is polynomially solvable for $K_{1,3}$-free graphs [18, 19], for bipartite graphs [19], for graphs with girth 5 at least [6], for graphs with a bounded maximal degree [3], for chordal graphs [15], and for graphs without cycles of lengths 4 and 5 [7]. It should be emphasized that the forbidden cycles are not necessarily induced.
For every graph $G$, the set of weight functions $w$ for which $G$ is $w$-well-covered is a vector space \[^3\]. That vector space is denoted $WCW(G)$ \[^2\]. Since recognizing well-covered graphs is \textbf{co-NP}-complete, finding the vector space $WCW(G)$ of an input graph $G$ is \textbf{co-NP}-hard. However, finding $WCW(G)$ can be done in polynomial time when the input is restricted to $K_{1,3}$-free graphs \[^{11}\], to graphs with a bounded maximal degree \[^3\], to graphs without cycles of lengths 4, 5 and 6 \[^{12}\], and to chordal graphs \[^1\].

### 1.3 Generating Subgraphs and Relating Edges

Further we make use of the following notions, which have been introduced in \[^9\]. Let $B$ be an induced complete bipartite subgraph of $G$ on vertex sets of bipartition $B_X$ and $B_Y$. Assume that there exists an independent set $S$ such that each of $S \cup B_X$ and $S \cup B_Y$ is a maximal independent set of $G$. Then $B$ is a \textit{generating} subgraph of $G$, and the set $S$ is a \textit{witness} that $B$ is generating. We observe that every weight function $w$ such that $G$ is $w$-well-covered must satisfy the restriction $w(B_X) = w(B_Y)$.

In the restricted case that the generating subgraph $B$ is isomorphic to $K_{1,1}$, call its vertices $x$ and $y$. In that case $xy$ is a \textit{relating} edge, and $w(x) = w(y)$ for every weight function $w$ such that $G$ is $w$-well-covered.

Recognizing relating edges is known to be \textbf{NP}-complete \[^2\], and it remains \textbf{NP}-complete even when the input is restricted to graphs without cycles of lengths 4 and 5 \[^{10}\], and to bipartite graphs \[^{13}\]. Therefore, recognizing generating subgraphs is also \textbf{NP}-complete for these restricted cases. However, recognizing relating edges can be done in polynomial time if the input is restricted to graphs without cycles of lengths 4 and 6 \[^{10}\], to graphs without cycles of lengths 5 and 6 \[^{12}\], and to graphs with a bounded maximal degree \[^{13}\].

It is also known that recognizing generating subgraphs is \textbf{NP}-complete for graphs with girth at least 6 \[^{13}\], and for $K_{1,4}$-free graphs \[^{13}\]. However, the problem is a polynomial solvable when the input is restricted to graphs without cycles of lengths 4, 6 and 7 \[^{9}\], to graphs without cycles of lengths 4, 5 and 6 \[^{12}\], to graphs without cycles of lengths 5, 6 and 7 \[^{12}\], to claw-free graphs \[^{19}\], and to graphs with a bounded maximal degree \[^{13}\].

### 1.4 Main Results

This paper is a continuation of the research performed in \[^{13}\].

Two restricted cases of the well-known \textbf{SAT} problem are presented in \[^{13}\], and proved to be \textbf{NP}-complete. In Section 2 we use these results to prove that another restricted case of the \textbf{SAT} problem, called \textbf{DMSAT}, is \textbf{NP}-complete as well.

In Section 3 we prove \textbf{NP}-completeness of two restricted cases of the recognizing generating subgraphs problem. In Subsection 3.1 we consider the case in which the input graph $G$ does not contain cycles of lengths 3 and 5, and $B$ is
\( K_{1,2} \). We use the main result of Section 2 for this proof. In Subsection 3.2 we deal with bipartite graphs with girth at least 6.

In Section 4 we present a polynomial algorithm for recognizing generating subgraphs of bipartite graphs without cycles of length 6. For this family of graphs we also supply a polynomial algorithm which finds \( WCW(G) \).

Section 5 contains a polynomial algorithm which finds \( WCW(G) \) for graphs without cycles of lengths 3, 4, 5, and 7. Especially, the algorithm works for bipartite graphs with girth at least 6.

The following open question is presented in [13]. Does there exist a family of graphs for which recognizing generating subgraphs is a polynomial task, but finding \( WCW(G) \) is co-NP-hard. Although we still do not know the answer for this question, Subsection 3.2 and Section 5 together give a first known example for the opposite case: a family of graphs for which recognizing generating subgraphs is an NP-complete problem, but finding the vector space \( WCW(G) \) is a polynomial task.

2 Binary Variables

A binary variable is a variable whose value is either 0 or 1. If \( x \) is a binary variable, then its negation is denoted by \( \overline{x} \). Each of \( x \) and \( \overline{x} \) is called a literal. Let \( X = \{x_1, \ldots, x_n\} \) be a set of binary variables. A clause \( c \) over \( X \) is a set of literals belonging to \( \{x_1, \overline{x_1}, \ldots, x_n, \overline{x_n}\} \) such that \( c \) does not contain both a variable and its negation. A truth assignment is a function

\[
\Phi : \{x_1, \overline{x_1}, \ldots, x_n, \overline{x_n}\} \rightarrow \{0, 1\}
\]

such that

\[
\Phi(\overline{x}_i) = 1 - \Phi(x_i) \text{ for each } 1 \leq i \leq n.
\]

A truth assignment \( \Phi \) satisfies a clause \( c \) if \( c \) contains at least one literal \( l \) such that \( \Phi(l) = 1 \).

In [13] the following problems about binary variables are presented.

**MONOTONE SAT** problem [8], [13]:

Input: A set \( X \) of binary variables and two sets, \( C_1 \) and \( C_2 \), of clauses over \( X \), such that all literals of the clauses belonging to \( C_1 \) are variables, and all literals of clauses belonging to \( C_2 \) are negations of variables.

Question: Is there a truth assignment for \( X \), which satisfies all clauses of \( C = C_1 \cup C_2 \)?

**Theorem 1** [8], [13] The **MONOTONE SAT** problem is NP-complete.

**DSAT** problem [13]:

Input: A set \( X \) of binary variables and a set \( C \) of clauses over \( X \) such that the following holds:

- Every clause contains 2 or 3 literals.
• Every two clauses have at most one literal in common.

• If two clauses, \( c_1 \) and \( c_2 \), have a common literal \( l_1 \), then there does not exist a literal \( l_2 \) such that \( c_1 \) contains \( l_2 \) and \( c_2 \) contains \( \overline{l_2} \).

**Question**: Is there a truth assignment for \( X \) which satisfies all clauses of \( C \)?

**Theorem 2** [13] *The DSAT problem is NP-complete.*

In this paper, we define the following problem.

**DMSAT** problem:
*Input*: A set \( X \) of binary variables and two sets, \( C_1 \) and \( C_2 \), of clauses over \( X \), such that the following holds:

• All literals of the clauses belonging to \( C_1 \) are variables.

• All literals of the clauses belonging to \( C_2 \) are negations of variables.

• Every clause of \( C_1 \) contains 2 or 3 literals.

• Every clause of \( C_2 \) contains 2 literals.

• Every two clauses of \( C_1 \) have at most one literal in common.

• Every two clauses of \( C_2 \) are disjoint.

**Question**: Is there a truth assignment for \( X \) which satisfies all clauses of \( C = C_1 \cup C_2 \)?

**Theorem 3** *The DMSAT problem is NP-complete.*

**Proof.** Obviously, the **DMSAT** problem is **NP**. We prove its **NP**-completeness by showing a reduction from the **DSAT** problem. Let

\[
I_1 = (X = \{x_1, ..., x_n\}, C = \{c_1, ..., c_m\})
\]

be an instance of the **DSAT** problem. Define \( Z = \{x_1, ..., x_n, z_1, ..., z_n\} \), where \( z_1, ..., z_n \) are new variables. For every \( 1 \leq j \leq m \), let \( c'_j \) be the clause obtained from \( c_j \) by replacing \( x_i \) with \( z_i \) for each \( 1 \leq i \leq n \). Let \( C' = \{c'_1, ..., c'_m\} \). For each \( 1 \leq i \leq n \) define two new clauses, \( d_i = (x_i, z_i) \) and \( e_i = (\overline{x_i}, \overline{z_i}) \). Let \( D = \{d_1, ..., d_n\} \) and \( E = \{e_1, ..., e_n\} \). Define \( I_2 = (Z, C' \cup D \cup E) \).

We show that \( I_2 \) is an instance of the **DMSAT** problem. Obviously, all literals of \( C' \cup D \) are variables, and all literals of \( E \) are negations of variables. Moreover, every clause of \( C' \) contains 2 or 3 literals, and every clause of \( D \cup E \) contains 2 literals. The fact that there are no 2 clauses of \( C \) with 2 common literals implies that there are no 2 clauses of \( C' \) with 2 common literals. A clause in \( C' \) and a clause in \( D \) cannot have 2 common literals, since a clause in \( C \) can not contain both a variable and its negation. A clause in \( C' \cup D \) and a clause in \( E \) do not have common literals, since all literals of \( C' \cup D \) are variables and all literals of \( E \) are negations of variables. By definition of \( E \), if
i \neq j \text{ then } e_i = (x_i, x_j) \text{ and } e_j = (x_j, x_i) \text{ are disjoint. Hence, } I_2 \text{ is an instance of the DMSAT problem, see Example 4.} \text{ It remains to prove that } I_1 \text{ and } I_2 \text{ are equivalent.}

Assume that } I_1 \text{ is a positive instance of the DSAT problem. There exists a truth assignment } \Phi_1 : \{x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n\} \rightarrow \{0, 1\} \text{ which satisfies all clauses of } C. \text{ Extract } \Phi_1 \text{ to a truth assignment } \Phi_2 : \{x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n, z_1, \overline{z}_1, \ldots, z_n, \overline{z}_n\} \rightarrow \{0, 1\} \text{ by defining } \Phi_2(z_i) = 1 - \Phi_1(x_i) \text{ for each } 1 \leq i \leq n. \text{ Clearly, } \Phi_2 \text{ is a truth assignment which satisfies all clauses of } C' \cup D \cup E. \text{ Hence, } I_2 \text{ is a positive instance of the DMSAT problem.}

Assume } I_2 \text{ is a positive instance of the DMSAT problem. There exists a truth assignment } \Phi_2 : \{x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n, z_1, \overline{z}_1, \ldots, z_n, \overline{z}_n\} \rightarrow \{0, 1\} \text{ that satisfies all clauses of } C' \cup D \cup E. \text{ For every } 1 \leq i \leq n \text{ it holds that } \Phi_2(z_i) = 1 - \Phi_2(x_i), \text{ or otherwise one of } d_i \text{ and } e_i \text{ is not satisfied. Therefore, } I_1 \text{ is a positive instance of the DSAT problem.}

Example 4 The following contains an instance of the DSAT problem and an equivalent instance of the DMSAT problem.

\( I_1 = (X, C_1) \), where \( X = \{x_1, \ldots, x_9\} \) and \( C_1 = \{(x_1, \overline{x}_2, x_3), (x_1, x_6, x_4), (x_1, x_7, x_3), (x_3, \overline{x}_3, x_5), (x_2, \overline{x}_3, x_8), (x_2, x_3, \overline{x}_9), (x_3, x_8, \overline{x}_6), (x_3, x_9, \overline{x}_7), (x_3, \overline{x}_7), (x_4, x_8), (x_4, \overline{x}_8), (x_5, x_9), (x_5, \overline{x}_5)\}. \)

\( I_2 = (Z, C_2) \), where \( Z = \{x_1, \ldots, x_9, z_1, \ldots, z_9\} \) and \( C_2 = \{(x_1, z_2, x_3), (x_1, x_6, x_4), (x_1, \overline{x}_7, x_3), (x_3, z_4, x_5), (x_2, z_3, x_8), (z_1, z_4, x_9), (x_3, x_6), (x_3, z_6), (x_3, \overline{x}_7), (x_4, x_8), (x_4, \overline{x}_8), (x_5, x_9), (x_5, z_9), (x_1, z_1), (x_1, \overline{x}_7), (x_2, z_2), (x_2, \overline{x}_2), (x_3, z_3), (x_3, \overline{x}_3), (x_4, z_4), (x_4, \overline{x}_4), (x_5, z_5), (x_5, \overline{x}_5), (x_6, z_6), (x_6, \overline{x}_6), (x_7, z_7), (x_7, \overline{x}_7), (x_8, z_8), (x_8, \overline{x}_8), (x_9, z_9), (x_9, \overline{x}_9)\}. \)

3 NP-Complete Results for Recognizing Generating Subgraphs

The GS problem is defined as follows.

Input: A graph \( G \) and an induced subgraph \( B \).

Question: Is \( B \) generating?

The GS problem is known to be NP-complete. In this section we prove that it remains NP-complete for two restricted cases: bipartite graphs with girth at least 6, and graphs without cycles of lengths 3 and 5.
3.1 Bipartite Graphs with Girth at Least 6

**Theorem 5** [13] The following problem is NP-complete.

*Input:* A graph $G$ with girth at least 6, and a subgraph $B$ of $G$.

*Question:* Is $B$ generating?

Theorem 5 is an instance of Theorem 6.

**Theorem 6** The following problem is NP-complete.

*Input:* A bipartite graph $G$ with girth at least 6, and a subgraph $B$ of $G$.

*Question:* Is $B$ generating?

**Proof.** The problem is obviously NP. We prove NP-completeness by showing a reduction from the DMSAT problem. Let

$I_1 = (X = \{x_1, \ldots, x_n\}, C = C_1 \cup C_2)$

be an instance of the DMSAT problem, where $C_1 = \{c_1, \ldots, c_m\}$ and $C_2 = \{c'_1, \ldots, c'_m\}$ are sets of clauses. Every clause of $C_1$ contains 2 or 3 variables, and every clause of $C_2$ contains 2 negations of variables. Every 2 clauses of $C_1$ have at most one literal in common, and every two clauses of $C_2$ are disjoint. Define a graph $G$ as follows:

$V(G) = \{x\} \cup \{y_j : 1 \leq j \leq m\} \cup \{v_j : 1 \leq j \leq m\} \cup \{v'_j : 1 \leq j \leq m'\} \cup \{u_i : 1 \leq i \leq n\} \cup \{u'_i : 1 \leq i \leq n\}$

$E(G) = \{xy_j : 1 \leq j \leq m\} \cup \{y_jv_j : 1 \leq j \leq m\} \cup \{xv'_j : 1 \leq j \leq m'\} \cup \{v_ju_i : x_i \text{ appears in } c_j\} \cup \{v'_ju'_i : \overline{x_i} \text{ appears in } c'_j\} \cup \{u_iu'_i : 1 \leq i \leq n\}$

Clearly, $G$ is bipartite, and the vertex sets of its bipartition are

$\{u_i : 1 \leq i \leq n\} \cup \{y_j : 1 \leq j \leq m\} \cup \{v'_j : 1 \leq j \leq m'\}$

and

$\{v_j : 1 \leq j \leq m\} \cup \{x\} \cup \{u'_i : 1 \leq i \leq n\}$.

Since $C_1$ does not contain two clauses with common two literals, and the clauses of $C_2$ are pairwise disjoint, $G$ does not contain cycles of length 4. Hence, its girth is at least 6. Let $B = G[\{x\} \cup \{y_j : 1 \leq j \leq m\}]$, and let $I_2 = (G, B)$ be an instance of the GS problem. It remains to prove that $I_1$ and $I_2$ are equivalent.

Assume that $I_1$ is a positive instance of the DMSAT problem. Let

$\Phi : \{x_1, \overline{x_1}, \ldots, x_n, \overline{x_n}\} \rightarrow \{0, 1\}$

be a truth assignment which satisfies all clauses of $C$. Let

$S = \{u_i : \Phi(x_i) = 1\} \cup \{u'_i : \Phi(x_i) = 0\}$. 
Let $S$ be the set that contains exactly one of $u \in G$. Hence, $S$ is a witness that $B$ is a generating subgraph of $G$. Therefore, $I_2$ is positive.

On the other hand, assume that $I_2$ is a positive instance of the GS problem. Let $S$ be a witness of $B$. Since $S$ is a maximal independent set of $\{u_i : 1 \leq i \leq n\} \cup \{u'_i : 1 \leq i \leq n\}$, exactly one of $u_i$ and $u'_i$ belongs to $S$, for every $1 \leq i \leq n$. Let

$$\Phi : \{x_1, \overline{x_1}, ..., x_n, \overline{x_n}\} \rightarrow \{0, 1\}$$

be a truth assignment defined by: $\Phi(x_i) = 1 \iff u_i \in S$. The fact that $S$ dominates

$$\{v_j : 1 \leq j \leq m\} \cup \{v'_j : 1 \leq j \leq m\}$$

implies that all clauses of $C$ are satisfied by $\Phi$. Therefore, $I_1$ is a positive instance of the DMSAT problem. ■

**Example 7** The following are an instance of the DSAT problem, an equivalent instance of the DMSAT problem, and an equivalent instance of the GS problem.

$I_1 = \{(x_1, \overline{x_3}, x_5), (x_1, x_3, \overline{x_5}), (x_1, x_7, x_9), (x_3, \overline{x_7}, x_9)\}$

$I_2 = \{(x_1, x_4, x_5), (x_2, x_3, x_6), (x_2, x_7, x_9), (x_3, x_8, x_{10}), (x_1, x_2), (\overline{x_1}, x_2), (x_3, x_4), (\overline{x_3}, x_4), (x_5, x_6), (\overline{x_5}, x_6), (x_7, x_8), (\overline{x_7}, x_8), (x_9, x_{10}), (x_9, \overline{x_{10}})\}$

$I_3 = (G, B)$, where $G$ is the graph shown in Figure 1 $B = G[\{x\} \cup \{y_j : 1 \leq j \leq 9\}]$.

The instance $I_2$ is positive because $x_1 = x_3 = x_6 = x_7 = x_{10} = 1$, $x_2 = x_4 = x_5 = x_8 = x_9 = 0$ is a satisfying assignment. The corresponding witness that $I_3$ is positive is the set $\{u_1, u'_2, u_3, u'_4, u'_5, u_6, u_7, u'_8, u'_9, u_{10}\}$.

### 3.2 Graphs Without Cycles of Lengths 3 and 5

**Theorem 8** [13] The GS problem is NP-complete even in the restricted case that $G$ is bipartite, and $B$ is $K_{1,1}$.

**Theorem 9** The GS problem is NP-complete even in the restricted case that $G$ does not contain cycles of lengths 3 and 5, and $B$ is $K_{1,2}$.

**Proof.** The problem is obviously in NP. We prove NP-completeness by showing a reduction from the MONOTONE SAT problem. Let

$$I_1 = (X = \{x_1, ..., x_n\}, C = C_1 \cup C_2)$$

be an instance of the MONOTONE SAT problem, where $X = \{x_1, ..., x_n\}$ is a set of 0–1 variables, $C_1 = \{c_1, ..., c_m\}$ is a set of clauses over the literals $\{x_1, ..., x_n\}$, and $C_2 = \{c'_1, ..., c'_{m'}\}$ is a set of clauses over the literals $\{\overline{x_1}, ..., \overline{x_n}\}$.  

8
Define a graph $G$ as follows:

$$V(G) = \{z, y_1, y_2\} \cup \{v_j : 1 \leq j \leq m\} \cup \{v'_j : 1 \leq j \leq m'\} \cup \{u_i : 1 \leq i \leq n\} \cup \{u'_i : 1 \leq i \leq n\},$$

$$E(G) = \{zy_1, zy_2\} \cup \{y_1v_j : 1 \leq j \leq m\} \cup \{y_2v'_j : 1 \leq j \leq m'\} \cup \{v_ju_i : x_i \text{ appears in } c_j\} \cup \{v'_j\overline{u'_i} : \overline{x_i} \text{ appears in } c'_j\} \cup \{u_iu'_i : 1 \leq i \leq n\}.$$}

The next step is proving that $G$ does not contain cycles of lengths 3 and 5.

If the edges $\{u_iu'_i : 1 \leq i \leq n\}$ are deleted from $G$, then the graph becomes bipartite, with the vertex sets of bipartition:

$$\{u_i : 1 \leq i \leq n\} \cup \{u'_i : 1 \leq i \leq n\} \cup \{y_1, y_2\}.$$
and
$$\{v_j : 1 \leq j \leq m\} \cup \{v'_j : 1 \leq j \leq m'\} \cup \{z\}.$$ 

Hence, it is enough to prove that an edge $u_iu'_i$ is not a part of a cycle of length 3 or 5. The vertices $u_i$ and $u'_i$ have no common neighbors. Therefore, $u_iu'_i$ is not a part of a triangle. Let $v_j \in N(u_i) \setminus \{u'_i\}$ and $v'_j \in N(u'_i) \setminus \{u_i\}$. Since $v_j$ and $v'_j$ have no common neighbors, $u_iu'_i$ is not a part of a cycle of length 5.

Let $B = G[\{z, y_1, y_2\}]$, and let $I_2 = (G, B)$ be an instance of the GS problem. It remains to prove that $I_1$ and $I_2$ are equivalent.

Assume that $I_1$ is a positive instance of the MONOTONE SAT problem. Let
$$\Phi : \{x_1, \overline{x_1}, \ldots, x_n, \overline{x_n}\} \rightarrow \{0, 1\}$$

be a truth assignment which satisfies all clauses of $C$. Let 
$$S = \{u_i : \Phi(x_i) = 1\} \cup \{u'_i : \Phi(x_i) = 0\}.$$ 

Obviously, $S$ is independent. Since $\Phi$ satisfies all clauses of $C$, every vertex of 
$$\{v_j : 1 \leq j \leq m\} \cup \{v'_j : 1 \leq j \leq m'\}$$

is adjacent to a vertex of $S$. Hence, $S$ is a witness that $B$ is a generating subgraph of $G$. Therefore, $I_2$ is positive.

On the other hand, assume that $I_2$ is a positive instance of the GS problem. Let $S$ be a witness of $B$. Since $S$ is a maximal independent set of 
$$\{u_i : 1 \leq i \leq n\} \cup \{u'_i : 1 \leq i \leq n\},$$

exactly one of $u_i$ and $u'_i$ belongs to $S$, for every $1 \leq i \leq n$. Let 
$$\Phi : \{x_1, \overline{x_1}, \ldots, x_n, \overline{x_n}\} \rightarrow \{0, 1\}$$

be a truth assignment defined by: $\Phi(x_i) = 1 \iff u_i \in S$. The fact that $S$ dominates 
$$\{v_j : 1 \leq j \leq m\} \cup \{v'_j : 1 \leq j \leq m'\}$$

implies that all clauses of $C$ are satisfied by $\Phi$. Therefore, $I_1$ is a positive instance of the MONOTONE SAT problem. $\blacksquare$

**Corollary 10** Let $p \geq 1$ and $q \geq 2$. The GS problem is NP-complete even in the restricted case that $G$ does not contain cycles of lengths 3 and 5, and $B$ is $K_{p,q}$.

**Proof.** We prove NP-completeness by showing a reduction from the MONOTONE SAT problem. Let $I$ be an instance of the MONOTONE SAT problem. Let $G$ be the graph constructed in the proof of Theorem 9 and contains the vertices $y_1, y_2, z$. Define $z_1 = z$, and let $H$ be the graph obtained from $G$ by adding vertices $y_3, \ldots, y_p, z_2, \ldots, z_q$ and edges $\{y_i z_j : 1 \leq i \leq p, 1 \leq j \leq q\}$. The following conditions are equivalent:
• \( I \) is a positive instance of the **MONOTONE SAT** problem.
• The induced subgraph of \( G \) with vertices \( y_1, y_2, z_1 \) is generating.
• The induced subgraph of \( H \) with vertices \( y_1, \ldots, y_p, z_1, \ldots, z_q \) is generating.

### 4 Bipartite Graphs Without Cycles of Length 6

In this section we present efficient algorithms for recognizing generating subgraphs, recognizing well-covered graphs, and finding the vector space \( WCW(G) \), in the restricted case that the input graph \( G \) is bipartite without cycles of length 6.

**Lemma 11** Let \( G \) be a bipartite graph without cycles of length 6, and let \( B \) be a bipartite subgraph of \( G \). Then \( N_2(B) \) is independent.

**Proof.** Denote the vertex sets of bipartition of \( B \) by \( B_X \) and \( B_Y \). Since \( G \) is bipartite, \( N(B_X) \cap N(B_Y) = \emptyset \). Therefore,

\[
N_2(B) = (N_2(B_X) \cap N_3(B_Y)) \cup (N_3(B_X) \cap N_2(B_Y))
\]

All vertices of \( N_2(B_X) \cap N_3(B_Y) \) belong to the same vertex set of bipartition of \( G \). Therefore, this set is independent. Similarly, \( N_3(B_X) \cap N_2(B_Y) \) is independent as well. Assume on the contrary that there exist two adjacent vertices, \( x'' \in N_2(B_X) \cap N_3(B_Y) \) and \( y'' \in N_3(B_X) \cap N_2(B_Y) \). Hence, there exist vertices \( x' \in N(x'') \cap N(B_X) \cap N_2(B_Y) \) and \( y' \in N(y'') \cap N_3(B_X) \cap N(B_Y) \). Let \( x \in N(x') \cap B_X \) and \( y \in N(y') \cap B_Y \). There exists a cycle of length 6 in \( G \), \( (x'', x', x, y, y', y'') \), which is a contradiction. Consequently, \( N_2(B) \) is independent. \( \blacksquare \)

**Theorem 12** There exists an \( O(|V(G)|^2) \) algorithm which solves the following problem:

**Input:** A bipartite graph \( G \) without cycles of length 6, and an induced subgraph \( B \) of \( G \).

**Question:** Is \( B \) generating?

**Proof.** Let \( B \) be an induced complete bipartite subgraph of \( G \) on vertex sets of bipartition \( B_X \) and \( B_Y \). Since \( G \) is bipartite, \( N(B_X) \cap N(B_Y) = \emptyset \). Define

\[
D_1 = N(B_X \cup B_Y) = (N(B_X) \cap N_2(B_Y)) \cup (N_2(B_X) \cap N(B_Y)),
\]

and

\[
D_2 = N_2(B_X \cup B_Y) = (N_2(B_X) \cap N_3(B_Y)) \cup (N_3(B_X) \cap N_2(B_Y)).
\]

Clearly, \( B \) is generating if and only if there exists an independent set in \( D_2 \) which dominates \( D_1 \). However, by Lemma 11, \( D_2 \) is independent. Hence, \( B \) is
generating if and only if $D_2$ dominates $D_1$. The following algorithm makes that decision.

**Algorithm 1: Generating(G,B)**

1. $D \leftarrow \bigcup_{v \in B_X \cup B_Y} N(v)$
2. $D_1 \leftarrow D \setminus (B_X \cup B_Y)$
3. foreach $v \in D_1$ do
   4. if $N(v) \setminus D = \emptyset$ then
      5. return $FALSE$
6. return $TRUE$

**Complexity of Algorithm 1**: The graph is stored as a boolean matrix. Sets of vertices are stored as boolean arrays of length $|V(G)|$. Hence, deciding whether an element belongs to a set is done in $O(1)$ time, while basic operations on sets such as union and difference are implemented in $O(|V(G)|)$ time. Finding the set of neighbors of a vertex is completed in $O(1)$ time.

In Line 1 and Line 2 there are $O(|V(G)|)$ operations on sets, each takes $O(|V(G)|)$ time. Therefore, the complexity of Line 1 and Line 2 is $O(|V(G)|^2)$. The foreach loop at Line 3 has $O(|V(G)|)$ iterations. Hence, Line 4 is performed $O(|V(G)|)$ times. One run of Line 4 takes $O(|V(G)|)$ time for evaluating the if condition. Therefore, the total time Algorithm 1 spends in Line 4 is $O(|V(G)|^2)$. Thus the complexity of the whole algorithm is $O(|V(G)|^2)$ as well.

Note that also Theorem 8 of [12], deals with a restricted case in which a subgraph $B$ is generating if and only if $N_2(B)$ dominates $N(B_X)\Delta N(B_Y)$. However, in that restricted case $G$ does not contain cycles of lengths 5, 6 and 7. Algorithm 1 does not fit [12], since in [12] the set $N_2(B)$ is not necessarily independent.

If $G$ is a graph without cycles of length 6, and $B$ is a subgraph of $G$ on vertex sets of bipartition $B_X$ and $B_Y$, then $\min(|B_X|, |B_Y|) \leq 2$. In the remaining part of Section 4 the following notation is used. (See Figure 2.) The set of vertices $X$ is independent, and $1 \leq |X| \leq 2$. If $|X| = 1$, then $X = \{x\}$, otherwise $X = \{x_1, x_2\}$. Moreover, $Y = \{y_1, ..., y_k\} = \bigcap_{x \in X} N(x)$. For every $1 \leq i \leq k$ define $A_i = N(y_i) \setminus X$ and $Z_i = N_2(y_i) \cap N_3(X)$. More notation: $S = N(X) \setminus Y$ and $S' = N(S) \setminus X$. Note that if $|X| = 1$ then $S = S' = \emptyset$.

**Lemma 13** Let $G$ be a bipartite graph without cycles of length 6. Let $X$ be an independent set of vertices such that $1 \leq |X| \leq 2$, and let $Y = \{y_1, ..., y_k\} = \bigcap_{x \in X} N(x)$. Assume that $|X| \leq |Y|$ and $B = G[X \cup Y]$ is generating. Let $Y'$ be a nonempty subset of $Y$, such that $y_i \in Y \setminus Y'$ implies $A_i \neq \emptyset$, for every $1 \leq i \leq k$. Then $B' = G[X \cup Y']$ is generating.

**Proof.** By Lemma 11, $N_2(V(B))$ and $N_2(V(B'))$ are independent sets. It remains to prove that $N(V(B'))$ is dominated by $N_2(V(B'))$. 


Output: A maximal set $T$ such a set does not exist.

Figure 2: Notation used in this Section. The sets $A_2, A_4, Z_2, Z_3, Z_4$ are empty. Since $S'$ dominates $S$, and $Z_i$ dominates $A_i$ for each $i \in \{2, 4, 5, 6\}$, the subgraph $G\{x_1, x_2, y_2, y_4, y_5, y_6\}$ is generating.

Clearly, $N(V(B)) = S \cup (\bigcup_{1 \leq i \leq k} A_i)$ and $N_2(V(B)) = S' \cup (\bigcup_{1 \leq i \leq k} Z_i)$. Since $B$ is generating, $N_2(V(B))$ dominates $N(V(B))$. Therefore, $S'$ dominates $S$, and $Z_i$ dominates $A_i$ for every $1 \leq i \leq k$.

Let $I = \{1 \leq i \leq k : y_i \in Y'\}$ and $\overline{I} = \{1 \leq i \leq k : y_i \not\in Y'\}$. Then $N(V(B')) = S \cup (Y' \setminus Y') \cup (\bigcup_{i \in I} A_i)$ and $N_2(V(B')) = S' \cup (\bigcup_{i \in I} A_i) \cup (\bigcup_{i \in \overline{I}} Z_i)$.

For every $i \in I$ it holds that $A_i \neq \emptyset$, and therefore $A_i \subseteq N_2(V(B'))$ dominates $y_i \in N(V(B'))$. For every $i \in I$ it holds that $Z_i \subseteq N_2(V(B'))$ dominates $A_i \subseteq N(V(B'))$. Hence, $N(V(B'))$ is dominated by $N_2(V(B'))$. ■

**Lemma 14** The following problem can be solved in $O(|V(G)|^3)$ time.

**Input:** A bipartite graph $G$ without cycles of length 6, and a vertex $x \in V(G)$.

**Output:** A maximal set $T \subseteq N(x)$ such that $G\{x \cup T\}$ is generating, or $\emptyset$ if such a set does not exist.

**Proof.** Let $N(x) = Y = \{y_1, ..., y_k\}$. Let $I = \{1 \leq i \leq k : A_i \subseteq N(Z_i)\}$ and $\overline{I} = \{1 \leq i \leq k : A_i \setminus N(Z_i) \neq \emptyset\}$. Let $T = \{y_i : i \in I\}$. Note that if $i \in \overline{I}$, then $A_i \neq \emptyset$. Each of the sets $\bigcup_{i=1}^{k} A_i$ and $\bigcup_{i=1}^{k} Z_i$ is independent, since all its vertices
belong to the same vertex set of bipartition of $G$.

Define $D_1 = N(\{x\} \cup T) = (Y \setminus T) \cup (\bigcup_{i \in I} A_i)$ and $D_2 = N_2(\{x\} \cup T) = (\bigcup_{i \in I} Z_i) \cup (\bigcup_{i \in I} A_i)$. By Lemma 11, $D_2$ is independent.

Assume $T \neq \emptyset$. We prove that $G[\{x\} \cup T]$ is generating. By Lemma 11, it is enough to prove that $D_1 \subseteq N(D_2)$. Let $1 \leq i \leq k$. If $i \in T$, then, by definition of $I$, $A_i \neq \emptyset$. Hence, $A_i \subseteq D_2$ dominates $y_i$. However, if $i \in I$, then $A_i \subseteq D_1$. Therefore, by definition of $I$, $A_i$ is dominated by $Z_i \subseteq D_2$.

On the other hand, let $B'$ be an induced complete bipartite subgraph of $G$ on vertex sets of bipartition $\{x\}$ and $T'$, such that there exists $y_i \in T' \setminus T$. There exists $a_i \in A_i$ which is not dominated by $Z_i$. Therefore, $N_2(\{x\} \cup T')$ does not dominate $N(\{x\} \cup T')$, and $B'$ is not generating. Hence, $T$ is a maximal set such that $G[\{x\} \cup T]$ is generating.

Let us conclude the proof by presenting the algorithm.

### Algorithm 2: MaxGen1($G, x$)

1. $T \leftarrow \emptyset$
2. foreach $y \in N(x)$ do
3. \hspace{1em} $A \leftarrow N(y) \setminus \{x\}$
4. \hspace{1em} $\text{flag} \leftarrow \text{TRUE}$
5. \hspace{1em} foreach $a \in A$ do
6. \hspace{2em} if $N(a) \setminus N(x) = \emptyset$ then
7. \hspace{3em} \hspace{1em} $\text{flag} \leftarrow \text{FALSE}$
8. \hspace{3em} \hspace{1em} break
9. \hspace{1em} if $\text{flag}$ then
10. \hspace{1em} \hspace{1em} $T \leftarrow T \cup \{y\}$
11. return $T$

**Complexity of Algorithm 2:** Each of the foreach loops in Lines 2 and 5 has $O(|V(G)|)$ iterations. Therefore, Lines 6-8 are performed $O(|V(G)|^2)$ times. The condition in Line 6 involves operations on sets. Hence, it takes $O(|V(G)|)$ time to evaluate it once. The total time Algorithm 2 spends in Line 6 is $O(|V(G)|^3)$. This is also the complexity of the algorithm. □

**Lemma 15** Let $G$ be a bipartite graph without cycles of length 6, and let $B$ be a generating subgraph of $G$ on vertex sets of bipartition $X$ and $Y = \{y_1, ..., y_k\}$, such that $k \geq 2$. Let $w \in WCW(G)$. Let $1 \leq j \leq k$ such that $A_j \neq \emptyset$. Then $w(y_j) = 0$.

**Proof.** The fact that $B$ is generating implies that $w(X) = w(Y)$.

Let $Y' = Y \setminus \{y_j\}$, and let $B' = G[X \cup Y']$. By Lemma 13, $B'$ is generating. Therefore, $w(X) = w(Y')$. Consequently, $w(Y - Y') = w(y_j) = 0$. □
Lemma 16 The following problem can be solved in $O(|V(G)|^2)$ time.
Input: A bipartite graph $G$ without cycles of length 6, and two vertices $x_1, x_2 \in V(G)$.
Output: A maximal set $T \subseteq N(x_1) \cap N(x_2)$ such that $|T| \geq 2$ and $G[\{x_1, x_2\} \cup T]$ is independent, or $\emptyset$ if such a set does not exist.

Proof. Let $X = \{x_1, x_2\}$ and $Y = N(x_1) \cap N(x_2) = \{y_1, \ldots, y_k\}$. Suppose $k \geq 2$. Let $S = N(X)\setminus Y$, and $S' = N(S)\setminus X$. (See Fig. 2.) If $S$ is not dominated by $S'$, then such a set $T$ does not exist, and the algorithm outputs $\emptyset$. Hence, assume $S \subseteq N(S')$.

Assume on the contrary that $(\bigcup_{1 \leq i \leq k} A_i) \cup S$ is not independent. There exist two adjacent vertices, $a \in \bigcup_{1 \leq i \leq k} A_i$ and $s \in S$. Assume without loss of generality that $s \in N(x_1)$. There exists $y \in Y \cap N(a)$, and $y' \in Y \setminus \{y\}$. Hence, $(a, y, x_2, y', x_1, s)$ is a cycle of length 6, which is a contradiction. Therefore, $(\bigcup_{1 \leq i \leq k} A_i) \cup S$ is independent. By Lemma 11, the set $S' \cup (\bigcup_{i=1}^k Z_i)$ is independent as well.

Let $T = \{y_i \in Y : A_i \subseteq N(Z_i)\}$. Note that if $y_i \notin T$, then $A_i \neq \emptyset$. Define
\[ D_1 = N(X \cup T) = \{y_i : y_i \notin T\} \cup \{A_i : y_i \in T\} \cup S \]
and
\[ D_2 = N_2(X \cup T) = \{Z_i : y_i \in T\} \cup \{A_i : y_i \notin T\} \cup S' \]

By Lemma 11, $D_2$ is independent.

Suppose $|T| \geq 2$. We prove that $G[X \cup T]$ is generating. By Lemma 11, it is enough to prove that $D_1 \subseteq N(D_2)$. We assume that $S \subseteq D_1$ is dominated by $S' \subseteq N(D_2)$. Let $1 \leq i \leq k$. If $y_i \in D_1$, then, by definition of $T$, $A_i \neq \emptyset$. Hence, $A_i \subseteq D_2$ dominates $y_i$. If $y_i \notin D_1$, then $A_i \subseteq D_1$. Therefore, by definition of $T$, $A_i$ is dominated by $Z_i \subseteq D_2$.

On the other hand, let $B'$ be an induced complete bipartite subgraph of $G$ on vertex sets of bipartition $X$ and $T'$, such that there exists $y_i \in T' \setminus T$. There exists $a_i \in A_i$ which is not dominated by $Z_i$. Therefore, $N_2(X \cup T')$ does not dominate $N(X \cup T')$, and $B'$ is not generating.

Let us conclude the proof by presenting the algorithm.
Algorithm 3: MaxGen2\((G, x_1, x_2)\)

1. \(T \leftarrow \emptyset\)
2. \(X \leftarrow \{x_1, x_2\}\)
3. \(Y \leftarrow N(x_1) \cap N(x_2)\)
4. if \(|Y| < 2\) then
5. \(\text{return } \emptyset\)
6. foreach \(s \in N(x_1) \Delta N(x_2)\) do
7. \(\text{if } N(s) \setminus X = \emptyset \text{ then } \text{return } \emptyset\)
8. foreach \(y \in Y\) do
9. \(\text{flag} \leftarrow \text{TRUE}\)
10. foreach \(a \in N(y) \setminus X\) do
11. \(\text{if } N(a) \setminus Y = \emptyset \text{ then } \text{flag} \leftarrow \text{FALSE}\)
12. \(\text{break}\)
13. if \(\text{flag} = \text{TRUE}\) then
14. \(T \leftarrow T \cup \{y\}\)
15. if \(|T| < 2\) then
16. \(\text{return } \emptyset\)
17. \(\text{return } T\)

Complexity of Algorithm 3 The complexity of Lines 1-5 is \(O(|V(G)|)\). The foreach loop in Line 6 has \(O(|V(G)|)\) iterations. In each iteration the condition of Line 7 is evaluated in \(O(|V(G)|)\) time. Therefore, the complexity of Lines 6-8 is \(O(|V(G)|^2)\).

If a vertex \(a \in \bigcup_{1 \leq i \leq k} A_i\) was adjacent to 3 distinct vertices, \(y, y', y''\) of \(Y\), then \(G\) contained a cycle of length 6, \((y, a, y', x_1, y'', x_2)\). Therefore, every \(a \in \bigcup_{1 \leq i \leq k} A_i\) is adjacent to two vertices of \(Y\) at most. Hence, the total number of iterations of the foreach loop of Line 11 is \(O(|V(G)|)\). Each evaluation of the condition of Line 12 takes \(O(|V(G)|)\) time, and the algorithm spends \(O(|V(G)|^2)\) time in Line 12. Hence, the complexity of Lines 9-16 is \(O(|V(G)|^2)\). The complexity of Lines 17-19 is \(O(|V(G)|)\). The total complexity of Algorithm 3 is \(O(|V(G)|^2)\).

Theorem 17 The following problem can be solved in \(O(|V(G)|^4)\) time.
Input: A bipartite graph \(G\) without cycles of length 6.
Output: WCW\((G)\).
Proof. The following algorithm receives as its input a bipartite graph \(G\) without cycles of length 6. The algorithm outputs a list of restrictions. For every function \(w : V(G) \rightarrow \mathbb{R}\) it holds that \(w \in WCW(G)\) if and only if \(w\) satisfies all restrictions outputted by the algorithm. The algorithm invokes the
MaxGen1 algorithm defined in the proof of Lemma \[14\] and the MaxGen2 algorithm defined in the proof of Lemma \[16\]

Algorithm 4: WCW(G)

```plaintext
1 foreach v ∈ V(G) do
2    T ←− MaxGen1(G, v)
3    if T ̸= ∅ then
4        output w(T) = w(v)
5    if |T| ≥ 2 then
6        foreach t ∈ T do
7            if N(t) ̸= {v} then
8                output w(t) = 0
9 foreach v1 ∈ V(G) do
10    foreach v2 ∈ V(G) \ {v1} do
11        T ←− MaxGen2(G, v1, v2)
12        if |T| ≥ 2 then
13            output w(T) = w({v1, v2})
14            foreach t ∈ T do
15                if N(t) ̸= {v1, v2} then
16                    output w(t) = 0
```

Complexity of Algorithm 4 The complexity of MaxGen1 is \(O(|V(G)|^3)\). That routine is invoked \(O(|V(G)|)\) times by this algorithm. The complexity of Lines 1-8 is \(O(|V(G)|^4)\). The complexity of MaxGen2 is \(O(|V(G)|^2)\). That routine is invoked \(O(|V(G)|^2)\) times by this algorithm. The complexity of Lines 9-16 is \(O(|V(G)|^4)\).

**Corollary 18** The following problem can be solved in \(O(|V(G)|^4)\) time. 
**Input:** A bipartite graph \(G\) without cycles of length 6.
**Question:** Is \(G\) well-covered?

**Proof.** In order to decide whether a graph \(G\) is well-covered, one can find the vector space \(WCW(G)\), and decide whether it includes the uniform function \(w ≡ 1\). However, we present an algorithm which is faster than Algorithm 4.
although it has the same computational complexity.

Algorithm 5: WC(G)

1. foreach $v \in V(G)$ do
2.   $T \leftarrow \text{MaxGen1}(G, v)$
3.   if $|T| > 1$ then
4.     return FALSE
5. foreach $v_1 \in V(G)$ do
6.   foreach $v_2 \in V(G) \setminus \{v_1\}$ do
7.     $T \leftarrow \text{MaxGen2}(G, v_1, v_2)$
8.     if $|T| > 1$ then
9.       return FALSE
10. if $|T| = 2$ then
11.   foreach $t \in T$ do
12.     if $N(t) \neq \{v_1, v_2\}$ then
13.       return FALSE
14. return TRUE

Complexity of Algorithm 5. The complexity of $\text{MaxGen1}$ is $O(|V(G)|^3)$, and it is invoked $O(|V(G)|)$ times by Algorithm 5. The complexity of $\text{MaxGen2}$ is $O(|V(G)|^2)$, and it is invoked $O(|V(G)|^2)$ times by Algorithm 5. The total complexity of Algorithm 5 is $O(|V(G)|^4)$. ■

5 Graphs Without Cycles of Lengths 3, 4, 5, 7

In this section, $G$ is a graph without cycles of lengths 3, 4, 5 and 7. Since $G$ does not contain small odd cycles, $G[N_3(v)]$ is bipartite for every $v \in V(G)$. Define $L(G) = \{v \in V(G) \mid d(v) = 1\}$, and $S_x = N(x) \setminus N(L(G))$ for every $x \in V(G) \setminus L(G)$. For every $y \in S_x$ it holds that $N(y) \cap N_2(x) \cap L(G) = \emptyset$. Therefore, $N(y) \cap N_2(x) \subseteq N(N_2(y) \cap N_3(x))$.

The main result of this section is a polynomial characterization of the vector space $WCW(G)$ for graphs without cycles of lengths 3, 4, 5 and 7.

Lemma 19 Let $G$ be a graph without cycles of lengths 3, 4, 5 and 7, and let $x \in V(G) \setminus L(G)$. Then $S_x$ is a maximal set with the following two properties:

1. $S_x \subseteq N(x)$
2. If $S_x \neq \emptyset$ then $G[\{x\} \cup S_x]$ is generating.

Proof. Obviously, $S_x \subseteq N(x)$. We assume $S_x \neq \emptyset$, and prove that $G[\{x\} \cup S_x]$ is generating. Let $T_1 = N_2(S_x) \cap N_3(x)$. Clearly, $T_1$ dominates $N(S_x) \cap N_2(x)$. Define $T_2 = N_2(x) \cap L(G)$. It holds that $T_2$ dominates $N(x) \setminus (S_x \cup L(G))$.
Since $T_1 \subseteq N_3(x)$, and $T_2 \subseteq N_2(x) \cap L(G)$, the set $T_1 \cup T_2$ is independent. Let $T$ be any maximal independent set of $G \setminus N[\{x\} \cup S_x]$, which contains $T_1 \cup T_2$. Obviously, $T$ is a witness that $G[\{x\} \cup S_x]$ is generating.

It remains to prove the maximality of $S_x$. Let $S'_x \subseteq N(x)$ such that there exists $y \in S'_x \setminus S_x$. We prove that $G[\{x\} \cup S'_x]$ is not generating. There exists \( l \in N(y) \cap L(G) \setminus N_2(x) \). Any independent set $S \subseteq V(G) \setminus N[\{x\} \cup S'_x]$ is not a witness that $G[\{x\} \cup S'_x]$ is generating, because it does not dominate $l$.

**Lemma 20** Let $G$ be a graph without cycles of lengths 3, 4, 5 and 7. Then $w(x) = w(L(G) \cap N(x))$ for every $x \in N(L(G))$ and for every $w \in WCW(G)$.

**Proof.** Let $x \in N(L(G))$ and let $T$ be a maximal independent set of $G \setminus N[x]$, which contains $!N_2(x)$. Then $T_1 = T \cup \{x\}$ and $T_2 = T \cup (N(x) \cap L(G))$ are maximal independent sets of $G$. The fact that $w(T_1) = w(T_2)$ implies $w(x) = w(L(G) \cap N(x))$.

**Lemma 21** Let $G$ be a graph without cycles of lengths 3, 4, 5 and 7. Then $w(y) = 0$ for every vertex $y \in V(G) \setminus N[L(G)]$ and for every $w \in WCW(G)$.

**Proof.** Let $x_1$ and $x_2$ be two distinct neighbors of $y$. Since $y \notin N[L(G)]$, it holds that $y \in S_{x_1}$ and $y \in S_{x_2}$. The fact that $G$ does not contain cycles of length 4 implies that $\{y\} = S_{x_1} \cap S_{x_2}$. By Lemma 19 each of $G[\{x_1\} \cup S_{x_1}]$ and $G[\{x_2\} \cup S_{x_2}]$ is generating. Therefore, $w(x_1) = w(S_{x_1})$ and $w(x_2) = w(S_{x_2})$, for every $w \in WCW(G)$.

Let $B = G[\{x_1, x_2\} \cup S_{x_1} \cup S_{x_2}]$. Clearly, $B$ is an induced bipartite subgraph of $G$, but it is not necessarily complete. Therefore, it does not fit the definition of a generating subgraph. However, we prove that there exists a set, $T^*$, such that $T^* \cup \{x_1, x_2\}$ and $T^* \cup S_{x_1} \cup S_{x_2}$ are maximal independent sets of $G$.

Let $T = (N_2(S_{x_1} \cup S_{x_2}) \cap N_3(\{x_1, x_2\})) \cup (N_2(\{x_1, x_2\}) \cap L(G))$. It is easy to see that $T$ is independent and dominates $N(B)$. Let $T^*$ be any maximal independent set of $G \setminus N[B]$ which contains $T$. The fact that $T^* \cup \{x_1, x_2\}$ and $T^* \cup S_{x_1} \cup S_{x_2}$ are maximal independent sets of $G$ implies that $w(\{x_1, x_2\}) = w(S_{x_1} \cup S_{x_2})$. Equivalently, $w(x_1) + w(x_2) = w(S_{x_1}) + w(S_{x_2}) - w(S_{x_1} \cap S_{x_2})$. Therefore, $w(y) = 0$.

**Theorem 22** Let $G$ be a connected graph without cycles of lengths 3, 4, 5 and 7, which is not isomorphic to $K_2$. Let $w : V(G) \rightarrow \mathbb{R}$. Then the following conditions are equivalent.

1. $w \in WCW(G)$
2. $w(x) = w(N(x) \cap L(G))$ for every vertex $x \in V(G) \setminus L(G)$.

**Proof.** If $G$ is an isolated vertex then $V(G) \setminus L(G) = \emptyset$, and every function defined on $V(G)$ belongs to $WCW(G)$. Therefore, Condition 1 and Condition 2 hold. In the remaining of the proof we assume that $|V(G)| \geq 3$.

Suppose that the first condition holds. By Lemmas 20 and 21 the second condition holds as well.
Assume the second condition holds. Denote \( N(L(G)) = \{v_1, ..., v_k\} \). We prove that the weight of every maximal independent set is \( \sum_{1 \leq i \leq k} w(v_i) \). For every \( 1 \leq i \leq k \) define \( T_i = \{v_i\} \cup (N(v_i) \cap L(G)) \). Let \( S \) be a maximal independent set of \( G \). For every \( 1 \leq i \leq k \) either \( S \cap T_i = \{v_i\} \) or \( S \cap T_i = N(v_i) \cap L(G) \). In both cases \( w(S \cap T_i) = w(v_i) \). Note that \( T_i \cap T_j = \emptyset \) for every \( 1 \leq i < j \leq k \).

Moreover, \( w(v) = 0 \) for every vertex \( v \in V(G) \setminus (\bigcup_{1 \leq i \leq k} T_i) \). Hence,

\[
    w(S) = \sum_{1 \leq i \leq k} w(S \cap T_i) + w(S \cap (V(G) \setminus (\bigcup_{1 \leq i \leq k} T_i)))
    = \sum_{1 \leq i \leq k} w(v_i) + 0 = w(N(L(G))).
\]

Therefore, the first condition holds.

Bipartite graphs with girth at least 6 are a restricted case of graphs without cycles of lengths 3, 4, 5, and 7. Hence, we obtain the following.

**Corollary 23** Let \( G \) be a connected bipartite graph with girth at least 6. Assume that \( G \) is not isomorphic to \( K_2 \). Let \( w : V(G) \rightarrow \mathbb{R} \). Then the following conditions are equivalent.

1. \( w \in WCW(G) \)
2. \( w(x) = w(N(x) \cap L(G)) \) for every vertex \( x \in V(G) \setminus L(G) \).

### 6 Conclusions and Future Work

In Subsection 3.2 we considered graphs without cycles of lengths 3 and 5. We proved that recognizing generating subgraphs isomorphic to \( K_{1,2} \) is an \( NP \)-complete task. By performing minor changes in the proof we showed that for every \( p \geq 1 \) and \( q \geq 2 \), recognizing generating subgraphs isomorphic to \( K_{p,q} \) is also \( NP \)-complete. Hence, we conjecture the following.

**Conjecture 24** Let \( i \leq p \) and \( j \leq q \). Let \( \Psi \) be a family of graphs for which recognizing generating subgraphs isomorphic to \( K_{i,j} \) is \( NP \)-complete. Then recognizing generating subgraphs of \( \Psi \) isomorphic to \( K_{p,q} \) is \( NP \)-complete as well.

We considered graphs which do not contain cycles of lengths 3 and 5. For this family of graphs we proved that recognizing generating subgraphs is an \( NP \)-complete problem. However, we still do not know the complexity statuses of recognizing relating edges, deciding whether a graph is well-covered, and finding the vector space \( WCW \).
References

[1] J. I. Brown, R. J. Nowakowski, *Well covered vector spaces of graphs*, SIAM Journal on Discrete Mathematics 19 (2006) 952–965.

[2] J. I. Brown, R. J. Nowakowski, I. E. Zverovich, *The structure of well-covered graphs with no cycles of length 4*, Discrete Mathematics 307 (2007) 2235–2245.

[3] Y. Caro, N. Ellingham, G. F. Ramey, *Local structure when all maximal independent sets have equal weight*, SIAM Journal on Discrete Mathematics 11 (1998) 644-654.

[4] Y. Caro, A. Sebő, M. Tarsi, *Recognizing greedy structures*, Journal of Algorithms 20 (1996) 137–156.

[5] V. Chvatal, P. J. Slater, *A note on well-covered graphs*, in: Quo Vadis, Graph Theory?, Annals of Discrete Mathematics 55, North Holland, Amsterdam, 1993, pp. 179–182.

[6] A. Finbow, B. Hartnell, R. Nowakowski, *A characterization of well-covered graphs of girth 5 or greater*, Journal of Combinatorial Theory B 57 (1993) 44–68.

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

[8] E. M. Gold, *Complexity of automaton identification from given data*, Inf. Control 37 (1978) 302–320.

[9] V. E. Levit, D. Tankus, *Weighted well-covered graphs without $C_4$, $C_5$, $C_6$, $C_7$*, Discrete Applied Mathematics 159 (2011) 354–359.

[10] V. E. Levit, D. Tankus, *On relating edges in graphs without cycles of length 4*, Journal of Discrete Algorithms 26 (2014) 28–33.

[11] V. E. Levit, D. Tankus, *Weighted well-covered claw-free graphs*, Discrete Mathematics 338 (2015) 99–106.

[12] V. E. Levit, D. Tankus, *Weighted well-covered graphs without cycles of lengths 4, 5 and 6*, Discrete Applied Mathematics 186 (2015) 158–167.

[13] V. E. Levit, D. Tankus, *Complexity results for generating subgraphs*, Algorithmica 80 (2018) 2384–2399.

[14] M. D. Plummer, *Some covering concepts in graphs*, Journal of Combinatorial Theory 8 (1970) 91–98.

[15] E. Prisner, J. Topp and P. D. Vestergaard, *Well-covered simplicial, chordal and circular arc graphs*, Journal of Graph Theory 21 (1996) 113–119.
[16] G. Ravindra, *Well-covered graphs*, Journal of Combinatorics, Information and System Sciences 2 (1977) 20-21.

[17] R. S. Sankaranarayana, L. K. Stewart, *Complexity results for well-covered graphs*, Networks 22 (1992) 247–262.

[18] D. Tankus, M. Tarsi, *Well-covered claw-free graphs*, Journal of Combinatorial Theory B 66 (1996) 293–302.

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