Recognizing Generating Subgraphs in Graphs without Cycles of Lengths 6 and 7

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

Abstract

Let B be an induced complete bipartite subgraph of G on vertex sets of bipartition B_X and B_Y. The subgraph B is *generating* if there exists an independent set S such that each of $S \cup B_X$ and $S \cup B_Y$ is a maximal independent set in the graph. If B is generating, it *produces* the restriction $w(B_X) = w(B_Y)$.

Let $w : V(G) \longrightarrow \mathbb{R}$ be a weight function. We say that G is w-well-covered if all maximal independent sets are of the same weight. The graph G is w-well-covered if and only if w satisfies all restrictions produced by all generating subgraphs of G. Therefore, generating subgraphs play an important role in characterizing weighted well-covered graphs.

It is an **NP**-complete problem to decide whether a subgraph is generating, even when the subgraph is isomorphic to $K_{1,1}$ [1]. We present a polynomial algorithm for recognizing generating subgraphs for graphs without cycles of lengths 6 and 7.

1 Introduction

1.1 Definitions and Notation

Throughout this paper $G = (V, E)$ is a simple (i.e., a finite, undirected, loopless and without multiple edges) graph with vertex set $V = V(G)$ and edge set $E = E(G)$.

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 $\mathcal{G}(\widehat{C_{i_1}}, .., \widehat{C_{i_k}})$ denote the family of all graphs which do not contain C_{i_1},...,C_{i_k}.

Let $S \subseteq V$ be a non-empty set of vertices, and let $i \in \mathbb{N}$. Then

$$N_i(S) = \{v \in V | \ min_{s \in S} \ d(v, s) = i\}, \ N_i[S] = \{v \in V | \ min_{s \in S} \ d(v, s) \leq i\}$$

where $d(x, y)$ is the minimal number of edges required to construct a path between x and y. If $i \neq j$ then $N_i(S) \cap N_j(S) = \emptyset$. We abbreviate $N_1(S)$ and $N_1[S]$ to $N(S)$ and $N[S]$, respectively. If $S = \{v\}$ for some $v \in V$, then $N_i(\{v\})$, $N_i[\{v\}]$, $N(\{v\})$, and $N[\{v\}]$, are abbreviated to $N_i(v)$, $N_i[v]$, $N(v)$, and $N[v]$, respectively.

A set of vertices $S \subseteq V$ is *independent* if for every $x, y \in S$, x and y are not adjacent. It is clear that an empty set is independent. An independent set is called *maximal* if it is not contained in another independent set. An independent set is *maximum* if the graph does not contain an independent set with a higher cardinality. A graph is called *well-covered* if every maximal independent set is maximum. The problem of finding a maximum cardinality independent set in an input graph is **NPC**. However, if the input is restricted to well-covered graphs, then a maximum cardinality independent set can be found polynomially using the *greedy algorithm*.

Let $T \subseteq V$. Then S *dominates* T if $T \subseteq N[S]$. If S and T are both empty, then $N(S) = \emptyset$, and S dominates T. If S is a maximal independent set of G, then it dominates the whole graph.

Let $w : V \longrightarrow \mathbb{R}$ be a weight function defined on the vertices of G. For every set $S \subseteq V$, define $w(S) = \Sigma_{s \in S} w(s)$. Then G is *w-well-covered* if all maximal independent sets of G are of the same weight. The set of weight functions w for which G is w-well-covered is a *vector space* [2].

The recognition of well-covered graphs is known to be **co-NP**-complete. This was proved independently in [4] and [12]. 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 [13, 14], for graphs with girth at least 5 [5], for graphs that contain neither 4- nor 5-cycles [6], for graphs with a bounded maximal degree [2], and for chordal graphs [11].

Recently, Levit and Tankus constructed a polynomial time algorithm for finding the vector space of weight functions w such that the input graph $G \in \mathcal{G}(\widehat{C_4}, \widehat{C_5}, \widehat{C_6})$ is w-well-covered [9]. They used the following notion. 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 called a *generating* subgraph of G, and it *produces* the restriction: $w(B_X) = w(B_Y)$. The set S is called a *witness* that B is generating. A graph G is w-well-covered for a weight function $w : V(G) \longrightarrow \mathbb{R}$ if and only if w satisfies all restrictions produced by all generating subgraphs of G. Therefore, generating subgraphs play an important role in characterizing w-well-covered graphs.

In the restricted case that the generating subgraph B is isomorphic to $K_{1,1}$, call its vertices x and y. In that case x and y are said to be *related*, xy is a *relating* edge, and $w(x) = w(y)$ for every weight function w such that G is w-well-covered. The witness of the related vertices x and y is an independent set S, containing neither x nor y, such that both $S \cup \{x\}$ and $S \cup \{y\}$ are maximal independent sets in the graph.

The decision problem whether an edge in an input graph is relating is **NP-**complete [1], and it remains **NP-**complete even when the input is restricted to graphs without cycles of lengths 4 and 5 [8] or to bipartite graphs [10]. Therefore, recognizing generating subgraphs is also **NP-**complete in these 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 [8], and to graphs without cycles of lengths 5 and 6 [9].

Recognizing generating subgraphs is **NP-**complete when the input is restricted to $K_{1,4}$-free graphs [10] or to graphs with girth at least 6 [10]. However, the problem is polynomial solvable when the input is restricted to graphs without cycles of lengths 4, 6 and 7 [7], to graphs without cycles of lengths 4, 5 and 6 [9], and to graphs without cycles of lengths 5, 6 and 7 [9].

1.2 Main Results

The subject of this paper is graphs without cycles of lengths 6 and 7. In Section 2 we define *extendable vertices*, and present a polynomial algorithm for recognizing extendable vertices in graphs without cycles of lengths 6 and 7.

Theorem 1.1 *The following problem can be solved in polynomial time:*
Input: A graph $G \in \mathcal{G}(\widehat{C}_6, \widehat{C}_7)$*, and a vertex* $x \in V(G)$*.*
Question: Is x *extendable?*

In Section 3 we use Theorem 1.1 to prove Theorem 1.2.

Theorem 1.2 *The following problem can be solved in polynomial time:*
Input: A graph $G \in \mathcal{G}(\widehat{C}_6, \widehat{C}_7)$*, and an edge* $xy \in E(G)$*.*
Question: Is xy *a relating edge?*

It is proved in [7] and [9] that recognizing generating subgraphs can be done polynomially for $\mathcal{G}(\widehat{C}_4, \widehat{C}_6, \widehat{C}_7)$, and $\mathcal{G}(\widehat{C}_5, \widehat{C}_6, \widehat{C}_7)$.

Theorem 1.3 *[7] The following problem can be solved in polynomial time:*
Input: A graph $G \in \mathcal{G}(\widehat{C}_4, \widehat{C}_6, \widehat{C}_7)$*, and an induced complete bipartite subgraph* B*.*
Question: Is B *generating?*

Theorem 1.4 *[9] The following problem can be solved in polynomial time:*
Input: A graph $G \in \mathcal{G}(\widehat{C}_5, \widehat{C}_6, \widehat{C}_7)$*, and an induced complete bipartite subgraph* B*.*
Question: Is B *generating?*

Theorem 1.3 and Theorem 1.4 are instances of Theorem 1.5, which is the main result of Section 4.

Theorem 1.5 *The following problem can be solved in polynomial time:*
Input: A graph $G \in \mathcal{G}(\widehat{C}_6, \widehat{C}_7)$*, and an induced complete bipartite subgraph* B*.*
Question: Is B *generating?*

A relating edge is a restricted case of a genereting subgraph. However, the complexity of the algorithm for recognizing related edges, presented in Section 3, is $O\left(|V|\left(|V|+|E|\right)\right)$, while the complexity of the algorithm which recognizes generating subgraphs in Section 4 is $O\left(|V|^2\left(|V|+|E|\right)\right)$.

2 Extendable Vertices

An *extendable* vertex is a vertex $v \in V(G)$ such that there does not exist an independent set in $N_2(v)$ which dominates $N(v)$. This notion was first introduced in [5]. If v is not extendable, a *witness* for non-extendability is a an independent set $S \subseteq N_2(v)$ such that $N(v) \subseteq N[S]$.

The main result of this section is a polynomial time algorithm which solves the following problem:
Input: A graph $G \in \mathcal{G}(\widehat{C}_6, \widehat{C}_7)$ and a vertex $x \in V(G)$.
Question: Is x extendable?

2.1 Graphs Without Cycles of length 6

In this subsection G is a graph without cycles of length 6, and x is a vertex in the graph, i.e. $G \in \mathcal{G}(\widehat{C}_6)$ and $x \in V(G)$.

Lemma 2.1 *Let (a, b, c) be a path in $G[N_2(x)]$. Then there exists a vertex, v, such that $\{v\} = N(x) \cap N(a) = N(x) \cap N(c)$.*

Proof. Assume on the contrary that Lemma 2.1 does not hold. Then there exist two distinct vertices, v_1 and v_2, such that $v_1 \in N(x) \cap N(a)$ and $v_2 \in N(x) \cap N(c)$. Therefore, (x, v_1, a, b, c, v_2) is a cycle of length 6, which is a contradiction. ∎

Lemma 2.2 *Let $P = (a_1, ..., a_{2k+1})$, $k \geq 1$, be a path (not necessarily simple) in $G[N_2(x)]$. Then there exists a vertex, v, such that $\{v\} = N(x) \cap N(a_1) = N(x) \cap N(a_{2k+1})$.*

Proof. By induction on k. If $k = 1$ then Lemma 2.2 is equivalent to Lemma 2.1.

Assume by induction that Lemma 2.2 holds for k. We prove that it holds also for $k+1$. Let $P = (a_1, ..., a_{2k+3})$ be a path in $G[N_2(x)]$. By the induction hypothesis, there exists a vertex, v, such that $\{v\} = N(x) \cap N(a_1) = N(x) \cap N(a_{2k+1})$. Considering the subpath $P' = (a_{2k+1}, a_{2k+2}, a_{2k+3})$, Lemma 2.1 implies that $N(x) \cap N(a_{2k+1}) = N(x) \cap N(a_{2k+3})$. Therefore, $\{v\} = N(x) \cap N(a_1) = N(x) \cap N(a_{2k+3})$. ∎

Lemma 2.3 *Let A be a connected component of $G[N_2(x)]$ which contains an odd cycle. Then $|N(x) \cap N(V(A))| = 1$.*

Proof. Let a be a vertex belonging to an odd cycle on A. For every vertex b in A, there exists a path P_b in A with even number of edges connecting a and b. By Lemma 2.2, there exists a vertex, v, such that $\{v\} = N(x) \cap N(a) = N(x) \cap N(b)$. Hence, $\{v\} = N(x) \cap N(V(A))$. ∎

Lemma 2.4 *Let A be a bipartite connected component of $G[N_2(x)]$ with vertex sets of bipartition V_1 and V_2. Then for each $1 \leq i \leq 2$, if $|V_i| \geq 2$ then $|N(x) \cap N(V_i)| = 1$.*

Proof. Let $1 \leq i \leq 2$ and $a \in V_i$. For every vertex $a \neq a' \in V_i$, there exists a path in A with even number of edges connecting a and a'. By Lemma 2.2, there exists a vertex, v, such that $\{v\} = N(x) \cap N(a) = N(x) \cap N(a')$. Therefore, $\{v\} = N(x) \cap N(V_i)$. ∎

Lemma 2.5 *Let A be a bipartite connected component of $G[N_2(x)]$ with vertex sets of bipartition V_1 and V_2, such that $min(|V_1|, |V_2|) \geq 2$. Then $|N(x) \cap N(V(A))| = 1$.*

Proof. A contains a path (a_1, a_2, a_3, a_4), where $a_1 \in V_1$ and $a_4 \in V_2$. By Lemma 2.4, for each $1 \leq i \leq 2$ there exists a vertex v_i, such that $\{v_i\} = N(x) \cap N(V_i)$. Assume on the contrary that $v_1 \neq v_2$. The cycle $(a_1, a_2, v_2, a_4, a_3, v_1)$ is of length 6, which is a contradiction. Therefore, $v_1 = v_2$, and $|N(x) \cap N(V(A))| = 1$. ∎

Corollary 2.6 *Let A be a connected component of $G[N_2(x)]$. Then at least one of the following options holds. (See Fig. 1.)*

1. $|V(A)| = 1$.
2. $|N(x) \cap N(V(A))| = 1$.

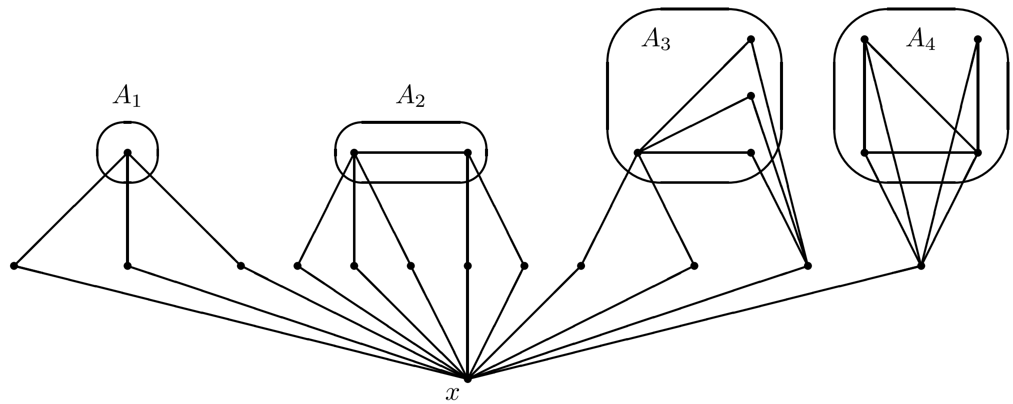

Figure 1: Distinct types of connected components of $N_2(x)$. A_1 contains only one vertex. A_4 contains an odd cycle, and therefore dominates only one vertex of $N(x)$. A_3 is bipartite with vertex sets of bipartition V_1 and V_2, $|V_1| = 1$, $|V_2| > 1$, and V_2 dominates only one vertex of $N(x)$. A_2 is $K_{1,1}$.

3. *A is $K_{1,r}$, for $r \geq 1$.*

Proof. Assume that the first 2 options of Corollary 2.6 do not hold for A. By Lemma 2.3, A is bipartite. By Lemma 2.5, at least one of the vertex sets of bipartition of A contains only one vertex. The third option of Corollary 2.6 holds. ■

Let A^* be the set of all connected components A of $G[N_2(x)]$ such that there exists a vertex $a \in V(A)$ for which $N(x) \cap N(a) = N(x) \cap N(V(A))$. For example, in Fig. 1, A_1 and A_4 belong to A^*, while A_2 and A_3 do not. Define $H_x = G[N_2[x] \setminus N[V(A^*)]]$. Since $G \in \mathcal{G}(\widehat{C}_6)$ and H_x is an induced subgraph of G, also $H_x \in \mathcal{G}(\widehat{C}_6)$.

Corollary 2.7 *In the graph H_x every connected component of $G[N_2(x)]$ is $K_{1,r}$ for $r \geq 1$.*

Proof. Follows immediately from the construction of H_x and Corollary 2.6. ■

Lemma 2.8 *The following conditions are equivalent.*

1. *x is not extendable in G.*

2. *x is not extendable in H_x.*

Proof.

$1 \implies 2$: Let $S \subseteq V(G)$ be an independent set in $N_2(x)$ which dominates $N(x)$. Let S' be the set of all vertices of S which exist also in the graph H_x, i.e. $S' = S \cap V(H_x)$. The set S' is a witness that x is not extendable in H_x.

$2 \implies 1$: Let $S' \subseteq V(H_x)$ be a witness that x is not extendable in H_x. For every component A of A^*, choose a vertex $v_A \in V(A)$ such that $N(x) \cap N(v_A) = N(x) \cap N(V(A))$. Define $S = S' \cup \{v_A | A \in A^*\}$. Then S is a witness that x is not extendable in G. ■

Lemma 2.9 *Let A be a connected component of $G[N_2(x)]$ in H_x, and a, b two adjacent vertices in A. Then the following conditions hold.*

1. $N(x) \cap N(V(A)) = N(x) \cap N(\{a, b\})$
2. $(N(a) \setminus N(b)) \cap N(x) \neq \emptyset$
3. $(N(b) \setminus N(a)) \cap N(x) \neq \emptyset$
4. $N(a) \cap N(b) \cap N(x) = \emptyset$

Proof. If there exists a vertex $z \in V(A) \setminus \{a, b\}$, then there exists a path with two edges connecting z to one of a or b. By Lemma 2.1, $N(x) \cap N(z) = N(x) \cap N(a)$ or $N(x) \cap N(z) = N(x) \cap N(b)$. Therefore, $N(x) \cap N(V(A)) = N(x) \cap N(\{a, b\})$.

If one of Conditions 2 and 3 does not hold, then there exists a vertex $v_A \in \{a, b\} \subseteq V(A)$ such that $N(x) \cap N(V(A)) = N(x) \cap N(v_A)$, which contradicts the construction of H_x. Therefore, Conditions 2 and 3 hold.

It follows from Conditions 2 and 3 that there exist vertices $v_a \in N(x) \cap (N(a) \setminus N(b))$ and $v_b \in N(x) \cap (N(b) \setminus N(a))$. If there existed a vertex $v \in N(a) \cap N(b) \cap N(x)$ then (x, v_a, a, v, b, v_b) was a cycle of length 6, which was a contradiction. Therefore, Condition 4 holds. ■

2.2 Graphs Without Cycles of Lengths 6 and 7

In this subsection $G \in \mathcal{G}(\widehat{C}_6, \widehat{C}_7)$, $x \in V(G)$, $H_x = G[N_2[x] \setminus N[V(A^*)]]$, and A is a connected component of $N_2(x)$ in H_x. The vertices a and b are adjacent to each other, and belong to A.

Lemma 2.10 *If $N(a) \cap N(x) = \{v_1, ..., v_k\}$ when $k > 2$, then $N(\{v_1, ..., v_k\}) \cap N_2(x) = \{a\}$.*

Proof. Corollary 2.7 implies that A is $K_{1,r}$, for $r \geq 1$. By Lemma 2.4, a is adjacent to all vertices of $V(A) \setminus \{a\}$. Assume on the contrary that Lemma 2.10 does not hold. There exists a vertex $a' \in (N(v_1) \cap N_2(x)) \setminus \{a\}$. Lemma 2.9 implies that $a' \notin V(A)$. Let A' be the connected component of $N_2(x)$ in H_x which contains a'. There exists a vertex $a'' \in V(A') \cap N(a')$.

Lemma 2.9 implies that there exists a vertex $v \in (N(a'') \setminus N(a')) \cap N(x)$. If $v \neq v_2$ then $(x, v, a'', a', v_1, a, v_2)$ is a cycle of length 7. Otherwise, $(x, v_2, a'', a', v_1, a, v_3)$ is a cycle of length 7. (See Fig. 2.) In both cases we obtained a contradiction. Therefore, Lemma 2.10 holds. ■

Lemma 2.11 *If $N(a) \cap N(x) = \{v_1, v_2\}$ then one of the following two options holds.*

- $N(\{v_1, v_2\}) \cap N_2(x) = \{a\}$.
- *There exists exactly one connected component $A' \neq A$ of $N_2(x)$ such that $\{v_1, v_2\} = N(V(A')) \cap N(x)$. For every connected component $A \neq A'' \neq A'$ of $N_2(x)$, it holds that $\{v_1, v_2\} \cap N(V(A'')) = \emptyset$.*

Proof. Clearly, A is $K_{1,r}$, for some $r \geq 1$, and a is adjacent to all vertices of $V(A) \setminus \{a\}$. By Lemma 2.9, there exists a vertex $v_b \in (N(b) \setminus N(a)) \cap N(x)$.

Assume that the first option of Lemma 2.11 does not hold. There exists a vertex $a' \in (N_2(x) \cap N(v_1)) \setminus \{a\}$. Lemma 2.9 implies that $a' \notin V(A)$. Let A' be the connected component of $N_2(x)$ which contains a', and let $a'' \in N_2(x) \cap N(a')$.

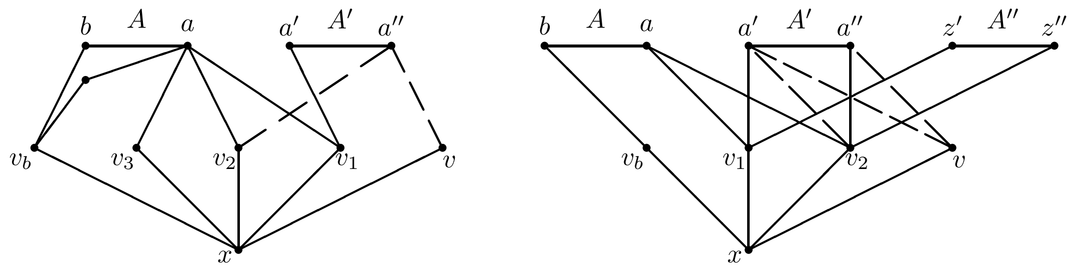

Figure 2: Proofs of Lemmas 2.10 (left) and 2.11 (right). The dashed edges are not in the graph.

We prove that $N(a'') \cap N(x) = \{v_2\}$. Lemma 2.9 implies that there exists a vertex $v \in (N(a'') \cap N(x)) \setminus \{v_1\}$. Assume on the contrary that $v \neq v_2$. Then $(x, v, a'', a', v_1, a, v_2)$ is a cycle of length 7, which is a contradiction. (See Fig. 2.) Therefore, $N(a'') \cap N(x) = \{v_2\}$.

We prove that $N(a') \cap N(x) = \{v_1\}$. By Lemma 2.9, $v_2 \notin N(a')$. If there exists a vertex $v \in (N(x) \cap N(a')) \setminus \{v_1, v_2\}$ then (x, v, a', v_1, a, v_2) is a cycle of length 6, which is a contradiction. Therefore, $N(a') \cap N(x) = \{v_1\}$.

Lemma 2.9 implies that $N(V(A')) \cap N(x) = N(\{a', a''\}) \cap N(x) = \{v_1, v_2\}$. Assume on the contrary that there exits a connected component $A \neq A'' \neq A'$ of $N_2(x)$ with vertices z' and z'', which are adjacent to v_1 and v_2, respectively. Then $(z', v_1, a', a'', v_2, z'')$ is a cycle of length 6, which is a contradiction. The second option of Lemma 2.11 holds. ■

Corollary 2.12 *If $|N(a) \cap N(x)| = 2$ then in the graph H_x every independent set of $N_2(x)$ which dominates $N(x)$, contains the vertex a.*

Proof. Denote $N(a) \cap N(x) = \{v_1, v_2\}$. Let S be a maximal independent set of $N_2(x)$ which dominates $N(x)$. If $N(\{v_1, v_2\}) \cap N_2(x) = \{a\}$ then obviously $a \in S$.

If $N(\{v_1, v_2\}) \cap N_2(x) \neq \{a\}$ then, by Lemma 2.11, there exits only one component $A' \neq A$ of $N_2(x)$ with vertices adjacent to $\{v_1, v_2\}$. However, no independent set in A' dominates both v_1 and v_2. Therefore, $a \in S$. ■

Corollary 2.13 *If $|N(a) \cap N(x)| = |N(b) \cap N(x)| = 2$ then in the graph H_x there does not exist an independent set of $N_2(x)$ which dominates $N(x)$.*

Proof. Suppose on the contrary that H_x contains an independent set $S \subseteq N_2(x)$ which dominates $N(x)$. By Corollary 2.12, S contains both a and b, which contradicts the fact that S is independent. ■

Let $S^* = \{v \in V(H_x) \mid v \in N_2(x),\ |N(v) \cap N(x)| > 1\}$. By Corollary 2.12, every independent set in $N_2(x)$ which dominates $N(x)$ contains S^*. Therefore, if S^* is not independent then there does not exist an independent set in $N_2(x)$ which dominates $N(x)$. In this case x is extendable in H_x and, by Lemma 2.8, x is also extendable in G.

Define H_x^* to be the induced subgraph of H_x with vertex set $V(H_x^*) = V(H_x) \setminus N[S^*]$.

Lemma 2.14 *Suppose S^* is independent. Then x is extendable in H_x if and only if it is extendable in H_x^*.*

Proof. Let $S_1 \subseteq V(H_x)$ be an independent set in $N_2(x)$ which dominates $N(x)$. Corollary 2.12 implies that $S^* \subseteq S_1$. Define $S_2 \subseteq V(H_x^*)$ by $S_2 = S_1 \setminus S^*$. Then in the graph H_x^* the set S_2 is independent, contained in $N_2(x)$, and dominates $N(x)$.

Let $S_2 \subseteq V(H_x^*)$ be an independent set in $N_2(x)$ which dominates $N(x)$. Define $S_1 \subseteq V(H_x)$ by $S_1 = S_2 \cup S^*$. Then S_1 is an independent set in $N_2(x)$ which dominates $N(x)$. ■

Theorem 2.15 *The following problem can be solved in* $O(|V|(|V|+|E|))$ *time.*
Input: A graph $G \in \mathcal{G}(\widehat{C}_6, \widehat{C}_7)$ *and a vertex* $x \in V(G)$.
Question: Is x extendable?

Proof. Let A be a connected component of $N_2(x)$ in the graph H_x^*. Then A is $K_{1,r}$ for some $r \geq 1$. Moreover, A is adjacent to exactly 2 vertices of $N(x)$, because otherwise the vertices of A were in $N[S^*]$, and not in $V(H_x^*)$. Every vertex set of bipartition of A dominates exactly one vertex of $N(x)$. In order to decide whether x is extendable in H_x^*, we define a flow network. We use the same technique as in Theorem 2.2 of [8].

Let $A_1, ..., A_k$ be the connected components of $N_2(x)$ in the graph H_x^*. Define a flow network $F_x = \{G_F = (V_F, E_F), s \in V_F, t \in V_F, c : E_F \longrightarrow \mathbb{R}\}$ as follows. (See Fig. 3.) Let $V_F = N_1(x) \cup N_2(x) \cup \{z_1, ..., z_k, s, t\}$, where $z_1, ..., z_k, s, t$ are new vertices, s and t are the source and sink of the network, respectively. The directed edges E_F are:

- the directed edges sz_i, for each $1 \leq i \leq k$;
- the directed edges $z_i a$, for each $1 \leq i \leq k$ and for each $a \in A_i$;
- all directed edges $v_2 v_1$ such that $v_2 \in N_2(x)$, $v_1 \in N(x)$ and $v_1 v_2 \in E(H_x^*)$;
- the directed edges vs, for each $v \in N(x)$;

Let $c \equiv 1$. Invoke any polynomial time algorithm for finding a maximum flow $f : E_F \longrightarrow \mathbb{R}$ in the network, for example Ford and Fulkerson's algorithm. The flow in a vertex $v \in V_F$ is defined by: $\Sigma_{(u,v) \in E_F} f(u,v)$.

Let S be the set of vertices in $N_2(x)$ in which there is a positive flow. It is easy to prove that S is independent, and every augmenting path in Ford and Fulkerson's algorithm increases by one each of $|f|$, $|S|$ and $|N(x) \cap N(S)|$. Therefore, $|f| = |S| = |N(x) \cap N(S)|$. Moreover, for every independent set S' of $N_2(x)$, it holds that $|N(x) \cap N(S)| \geq |N(x) \cap N(S')|$. For more details see [8].

If S dominates $N(x)$ in H_x^* then obviously x is not extendable. Otherwise, there does not exist an independent set in $N_2(x)$ which dominates $N(x)$, and therefore x is extendable.

The following polynomial algorithm receives as its input a graph $G \in \mathcal{G}(\widehat{C}_6, \widehat{C}_7)$ and a vertex $x \in V(G)$. If x is extendable, the algorithm returns $\emptyset$. Otherwise, the algorithm

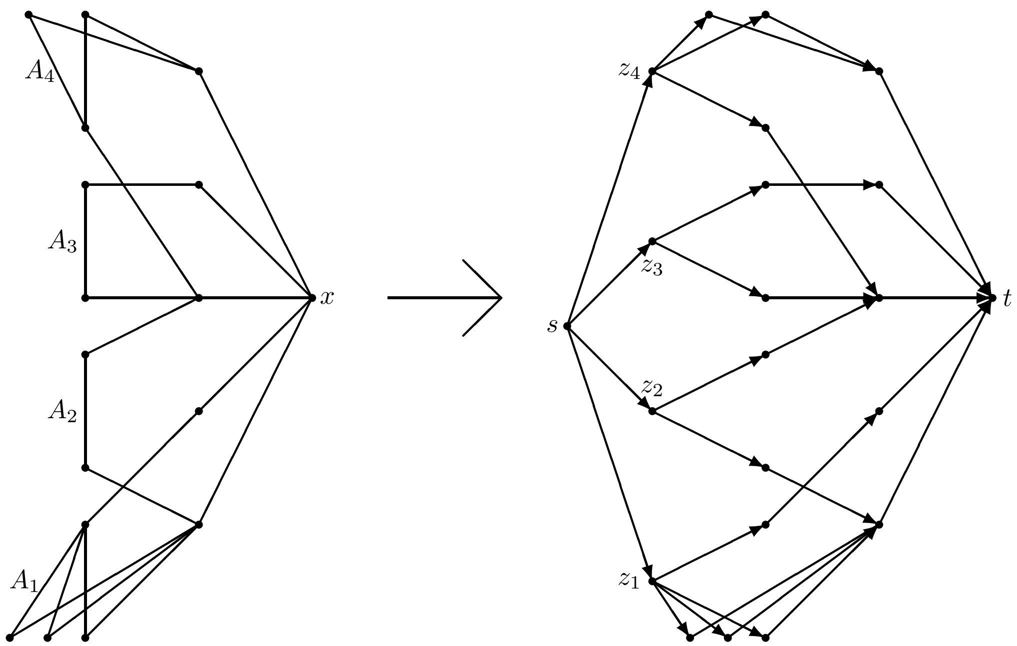

Figure 3: An example of the construction of the flow network F_x (right) from H_x^* (left).

returns a witness that x is not extendable.

Algorithm 1: Decide whether x is extendable in $G \in \mathcal{G}(\widehat{C}_6, \widehat{C}_7)$.

$T \longleftarrow \emptyset$
Find the connected components of $N_2(x)$
Find A^*
for each $A \in A^*$ **do**
  Choose $v_A \in V(A)$ such that $N(v_A) \cap N(x) = N(V(A)) \cap N(x)$
  $T \longleftarrow T \cup \{v_A\}$
Construct the graph H_x.
$S^* \longleftarrow \{a \in V(H_x) \mid a \in N_2(x), |N(a) \cap N(x)| > 1\}$.
if S^* *is not independent* **then**
  return $\emptyset$.
Construct the graph H_x^*.
Construct the flow network F_x
Find a maximum flow $f_x : E_F \longrightarrow \mathbb{R}$ in the network F_x.
Let S be the set of vertices in $N_2(x)$ in which there is a positive flow.
if S *does not dominate* $N(x)$ *in* H_x^* **then**
  return $\emptyset$.
return $T \cup S \cup S^*$.

Correctness of Algorithm 1: Follows from previous lemmas.

Complexity analysis of Algorithm 1: Constructing the set T includes finding $N_2(x)$, the connected components of $G[N_2(x)]$, and the set A^*. This can be implemented in $O(|V|+|E|)$ time. Also finding the induced subgraphs H_x can be implemented in $O(|V|+|E|)$ time. Finding S_x^* and deciding whether it is independent can be done in $O(|V|+|E|)$ time, as well. This is also the complexity for constructing the induced subgraph H_x^*, and the flow network F_x.

One iteration of Ford and Fulkerson's algorithm can be implemented in $O(|V|+|E|)$ time. In each iteration the number of vertices in $N_2(x)$ with a positive flow increases by 1. Therefore, the number of iterations can not exceed $|V|$, and Ford and Fulkerson's algorithm terminates in $O(|V|(|V|+|E|))$ time.

Deciding whether S_x dominates $N(x)$ can be done in $O(|V|+|E|)$ time. The total complexity of Algorithm 1 is $O(|V|(|V|+|E|))$. ■

3 Relating Edges

In this section we present a polynomial algorithm for recognizing relating edges in graphs without cycles of lengths 6 and 7.

Lemma 3.1 *Let $G \in \mathcal{G}(\widehat{C}_6, \widehat{C}_7)$ and $xy \in E(G)$. The following conditions are equivalent.*

1. *The edge xy is relating.*
2. *There exist an independent set, $S_x \subseteq N_2(x) \setminus N(y)$, which dominates $N(x) \cap N_2(y)$, and an independent set, $S_y \subseteq N_2(y) \setminus N(x)$, which dominates $N(y) \cap N_2(x)$.*

Proof. $1 \implies 2$: Let S be a witness that xy is relating. Define $S_x = S \cap N_2(x)$ and $S_y = S \cap N_2(y)$. Clearly, S_x dominates $N(x) \cap N_2(y)$ and S_y dominates $N(y) \cap N_2(x)$.

$2 \implies 1$: Let $x'' \in S_x$ and $y'' \in S_y$. There exist vertices $x' \in N(x) \cap N(x'')$ and $y' \in N(y) \cap N(y'')$. If x'' and y'' were adjacent then (x'', x', x, y, y', y'') was a cycle of length 6. Therefore, $S_x \cup S_y$ is independent. Let S be a maximal independent set of $G[V(G) \setminus N[\{x,y\}]]$ which contains $S_x \cup S_y$. Then S is a witness that xy is relating. ■

Theorem 3.2 *The following problem can be solved in $O(|V|(|V|+|E|))$ time.*
Input: A graph $G \in \mathcal{G}(\widehat{C}_6, \widehat{C}_7)$ and an edge $xy \in E(G)$.
Question: Is xy relating?

Proof. The following polynomial algorithm receives as its input a graph $G \in \mathcal{G}(\widehat{C}_6, \widehat{C}_7)$ and an edge $xy \in E(G)$. If xy is relating, the algorithm returns an independent set in $N_2(\{x,y\})$

which dominates $N(x)\Delta N(y)$. Otherwise, $\emptyset$ is returned.

Algorithm 2: Decide whether the edge xy in the graph $G \in \mathcal{G}(\widehat{C}_6, \widehat{C}_7)$ is relating.

$A \longleftarrow \emptyset$
$B \longleftarrow \emptyset$
if $N(x) \cap N_2(y) \neq \emptyset$ **then**
    $A \longleftarrow Alg1(G[(N_2(x) \cap N_3(y)) \cup (N(x) \cap N_2(y)) \cup \{x\}], x)$
    if $A = \emptyset$ **then**
        return $\emptyset$
if $N(y) \cap N_2(x) \neq \emptyset$ **then**
    $B \longleftarrow Alg1(G[(N_2(y) \cap N_3(x)) \cup (N(y) \cap N_2(x)) \cup \{y\}], y)$
    if $B = \emptyset$ **then**
        return $\emptyset$
return $A \cup B$.

Correctness of Algorithm 2: Follows from Lemma 3.1.

Complexity analysis of Algorithm 2: The algorithm invokes Algorithm 1 at most twice. Therefore, the complexity of Algorithm 2 is equal to the complexity of Algorithm 1, i.e. $O(|V|(|V| + |E|))$. ■

4 Generating Subgraphs

In this section we present a polynomial algorithm for recognizing generating subgraphs in $\mathcal{G}(\widehat{C}_6, \widehat{C}_7)$. Through this section $G \in \mathcal{G}(\widehat{C}_6, \widehat{C}_7)$, and B is an induced complete bipartite subgraph of G.

Lemma 4.1 *Let a and b be two distinct vertices in $V(B)$. Let $a' \in N(a) \setminus V(B)$ and $b' \in N(b) \setminus V(B)$ such that $a' \neq b'$. Let $a'' \in N_2(a) \cap N(a')$ and $b'' \in N_2(b) \cap N(b')$. Then a'' and b'' are not adjacent.*

Proof. If a and b are adjacent, the lemma holds, or otherwise (a'', a', a, b, b', b'') is a cycle of length 6.

If a and b are not neighbors, there exists a vertex $c \in V(B) \cap N(a) \cap N(b)$. If $a'' \in N(b'')$ then $(a'', a', a, c, b, b', b'')$ is a cycle of length 7, which is a contradiction. ■

For every vertex $b \in V(B)$, define $G_b = G[(N_2(b) \cap N_2(V(B))) \cup (N(b) \cap N(V(B))) \cup \{b\}]$. Since $G \in \mathcal{G}(\widehat{C}_6, \widehat{C}_7)$ and G_b is an induced subgraph of G, also $G_b \in \mathcal{G}(\widehat{C}_6, \widehat{C}_7)$.

Corollary 4.2 *Let a and b be two distinct vertices in $V(B)$. Let $S_a \subseteq V(G_a) \cap N_2(a)$ and $S_b \subseteq V(G_b) \cap N_2(b)$ be two independent sets. Then $S_a \cup S_b$ is an independent set in G.*

Lemma 4.3 *The following conditions are equivalent.*

1. *The subgraph B is generating in G.*
2. *The vertex b is not extendable in G_b, for every $b \in V(B)$.*

Proof.

1 $\implies$ 2: Let S be a witness that B is a generating subgraph of G. For every $b \in V(B)$ define $S_b = S \cap V(G_b)$. Then S_b is a witness that b is not extendable in G_b.

2 $\implies$ 1: For every $b \in V(B)$, let S_b be a witness that b is not extendable in G_b. Define $S = \bigcup_{b \in V(B)} S_b$. By Corollary 4.2, S is independent. The set S is a witness that B is a generating subgraph of G. ■

Theorem 4.4 *The following problem can be solved in* $O\left(|V|^2 (|V| + |E|)\right)$ *time.*
Input: A graph $G \in \mathcal{G}(\widehat{C}_6, \widehat{C}_7)$ *and an induced bipartite subgraph* B.
Question: Is B *generating?*

Proof. The following algorithm receives as its input a graph $G \in \mathcal{G}(\widehat{C}_6, \widehat{C}_7)$ and an induced complete bipartite subgraph, B. The algorithm returns a witness that B is generating, if exists, and $\emptyset$ otherwise.

Algorithm 3: Decide whether the subgraph B of $G \in \mathcal{G}(\widehat{C}_6, \widehat{C}_7)$ is generating.

---

for each $b \in V(B)$ **do**
  $S_b \longleftarrow Alg1(G_b, b)$
  if $S_b = \emptyset$ *and* $|V(B)| > 1$ **then**
    return $\emptyset$
$S \longleftarrow \bigcup_{b \in V(B)} S_b$.
Extract S to a maximal independent set, S^*, of $G \setminus N[V(B)]$.
return S^*.

Correctness of Algorithm 3: Follows from Lemma 4.3.

Complexity analysis of Algorithm 3: The complexity of Algorithm 1 is $O\left(|V| (|V| + |E|)\right)$, and it is invoked at most $O(|V|)$ times. Therefore, the total complexity of Algorithm 3 is $O\left(|V|^2 (|V| + |E|)\right)$. ■

5 Conclusions and Future Work

We presented a polynomial algorithm for recognizing generating subgraphs in $\mathcal{G}(\widehat{C}_6, \widehat{C}_7)$. However, we did not find a polynomial algorithm which receives $G \in \mathcal{G}(\widehat{C}_6, \widehat{C}_7)$ as its input, and finds $WCW(G)$, the vector space of all weight functions w such that G is w-well-covered. Checking all induced complete bipartite subgraphs is obviously not a polynomial algorithm, since the number of checked subgraphs can be exponential. Nevertheless, we conjecture the following.

Conjecture 5.1 *The following problem can be solved in polynomial time.*
Input: A graph $G \in \mathcal{G}(\widehat{C}_6, \widehat{C}_7)$.
Output: $WCW(G)$.

It is known that recognizing relating edges is a polynomial task for $\mathcal{G}(\widehat{C}_4, \widehat{C}_6)$ [8], and for $\mathcal{G}(\widehat{C}_5, \widehat{C}_6)$ [9]. We proved that also for $\mathcal{G}(\widehat{C}_6, \widehat{C}_7)$ the problem is polynomial. However, in most parts of the proof, the only forbidden cycles were of length 6. Hence, we conjecture that all three theorems are instances of the following.

Conjecture 5.2 *The following problem is polynomial solvable:*
Input: A graph $G \in \mathcal{G}(\widehat{C}_6)$ *and an edge* $xy \in E(G)$.
Question: Is xy *relating?*

References

[1] 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.

[2] 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.

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

[4] V. Chvatal, P. J. Slater, *A note on well-covered graphs*, Quo vadis, Graph Theory Ann Discr Math 55, North Holland, Amsterdam, 1993, 179-182.

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

[6] 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.

[7] 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.

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

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

[10] V. E. Levit, D. Tankus, *Complexity Results for Generating Subgraphs*, Algorithmica **80** (2018) 2384-2399.

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

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

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

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