k-ary spanning (Steiner) tree is hard

R. Mahendra Kumar[1], A. Mohanapriya[1], N. Rahul[1], N. Sadagopan[1], B. Surya Raghav[1], and J. Vibulan[1]

Indian Institute of Information Technology, Design and Manufacturing, Kancheepuram, India
{coe18d004,coe19d003,cs21b1082,sadagopan,cs21b2042,cs21b2043}@iiitdm.ac.in

Abstract. A tree T is said to be a k-ary tree if the degree of each vertex in T is at most $k+1$. For a connected graph G, the k-ary spanning tree problem (k-ary ST), $k \geq 1$ which asks for a k-ary tree T spanning $V(G)$. When $k=1$, the k-ary spanning tree problem reduces to the well-known Hamiltonian path problem. It is known that the Hamiltonian path problem is NP-complete on various well-known special graph classes. In this paper, we initiate the classical complexity study of the k-ary spanning tree problem, for each $k \geq 2$. We study the problem on well-known special graph classes, which are chordal bipartite graphs, strongly chordal graphs, and split graphs.

We prove that for any $k \geq 2$, k-ary ST is NP-complete on chordal bipartite graphs, whereas, for a subclass of chordal bipartite graphs which are bipartite chain graphs, the problem is polynomial time, for any $k \geq 2$. Further, for any $k \geq 2$, the computational complexity of the problem is NP-complete on strongly chordal split graphs. Finally, we prove that for any $k \geq 2$, the problem remains NP-complete on $K_{1,r}$-free split graphs, when $r \geq k+4$ and is polynomial-time solvable on $K_{1,r}$-free split graphs, when $r \leq k+2$.

Further, we extend our results of k-ST to the k-ary Steiner tree problem which is defined as follows: Given a connected graph G, and a subset $R \subseteq V(G)$, the k-ary Steiner tree problem (k-ary STREE), for $k \geq 2$, the objective is to find a set $S \subseteq (V(G) \setminus R)$ such that there exists a k-ary tree spanning $G[R \cup S]$. Interestingly, we prove that the computational complexity of k-ary ST and k-ary STREE is the same for the special graph classes considered.

Keywords: k-ary spanning tree · chordal bipartite graphs · strongly chordal split graphs

1 Introduction

The Hamiltonian path problem (HPATH), and The Steiner tree problem (STREE) remains to be ingenious problems in graph theory since its discovery. These problems are of great theoretical and practical importance. In this paper, we consider the generalization of these two problems. For a connected graph G, a Hamiltonian path P is a path spanning $V(G)$. The Hamiltonian path problem (HPATH) is a well-studied combinatorial problem. It is known [6], that HPATH is NP-complete on bipartite graphs, chordal bipartite graphs, strongly chordal graphs, and split graphs. A fine-grained complexity analysis of the problem on split graphs is reported in the literature [8], that is for $K_{1,5}$-free split graphs the problem is NP-complete whereas for $K_{1,4}$-free split graphs the problem is polynomial-time solvable. Further, it is known that HPATH is polynomial-time solvable on interval graphs [2], bipartite chain graphs, bipartite permutation graphs [3], and distance hereditary graphs [5].

The Hamiltonian path problem can be restated as *the unary spanning tree problem* which is defined as follows: *For a graph G, does there exist a unary tree spanning $V(G)$?* We generalize this notion of spanning tree to k-ary spanning tree, for $k \geq 2$. A tree T is said to be a k-ary tree if the degree of each vertex in T is at most $k+1$. The k-ary spanning tree problem (k-ary ST), $k \geq 2$ asks for a spanning tree T in G such that T is a k-ary spanning tree.

For $k \geq 2$, the k-ary spanning tree problem is a generalization of the unary spanning tree problem (HPATH). Observe that a unary spanning tree is also a k-ary spanning tree, for any $k \geq 2$. In general, the following observation is true.

Observation 1. *For $q \geq 1$, a q-ary spanning tree is also a k-ary spanning tree, where $k \geq q$.*

From Observation 1, it is clear that for an instance $\mathcal{I}$, if $\mathcal{I}$ is an yes-instance of HPATH, then $\mathcal{I}$ is also an yes-instance of k-ary ST, for any $k \geq 2$. But it is not true for the converse of Observation 1. Consider an instance $\mathcal{I}'$ which is a caterpillar graph with each vertex in the backbone path having k pendant vertices. Clearly, $\mathcal{I}'$ is a no-instance of HPATH, but it is an yes-instance of q-ary ST, for $q \geq k-1$. Similarly, $\mathcal{I}'$ is a no-instance for q-ary ST, for $q < k-1$. The boundary of an yes-instance of the k-ary spanning tree problem for each $k \geq 2$ is not the same (see Figure 1). Hence, analyzing the computational complexity of the k-ary spanning tree problem, for each $k \geq 2$ is necessary.

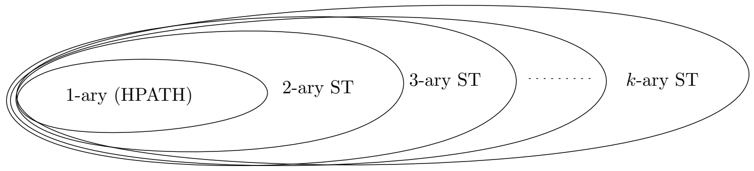

Fig. 1. A comparison about yes-instances of k-ary spanning tree, $\forall k \geq 1$.

Secondly, we consider the Steiner tree problem. The Steiner tree problem (STREE) can be defined as follows: given a connected graph $G = (V, E)$ and a subset of vertices $R \subseteq V(G)$, the objective is to find a minimum cardinality set $S \subseteq V(G)$ such that the set $R \cup S$ induces a connected subgraph. STREE is known to be NP-complete in split graphs, and chordal bipartite graphs[7]. Further, it is known that the problem is polynomial-time solvable on strongly chordal graphs, bipartite distance hereditary graphs, interval graphs, and convex bipartite graphs [9,4].

A vertex in the resultant tree from the Steiner tree problem can have an arbitrary degree. Now, we ask for the complexity of the problem when each vertex in the Steiner tree problem having degree at most $k \geq 2$, which can be precisely stated as the following k-ary Steiner tree problem (k-ary STREE) follows: Given a connected graph G, and a subset $R \subseteq V(G)$, the k-ary Steiner tree problem (k-ary STREE), for $k \geq 2$, the objective is to find a set $S \subseteq (V(G) \setminus R)$ such that there exists a k-ary tree spanning $G[R \cup S]$.

It is interesting to note that the classical Steiner tree problem is polynomial-time solvable on Strongly chordal graphs, whereas the k-ary Steiner tree problem on strongly chordal split graphs is NP-complete, for each $k \geq 2$. In this paper, we analyze the computational complexity of the k-ary Steiner tree problem (k-ary STREE), for each $k \geq 2$.

Our contributions. In this paper, we consider the computational complexity of the k-ary spanning tree problem (k-ary ST), and the k-Steiner tree problem (k-ary STREE). We prove the following results, for each $k \geq 2$.

1. We prove k-ary ST and k-ary STREE are NP-complete on chordal bipartite graphs in Section 2
2. We prove that k-ary ST and k-ary STREE are polynomial-time solvable on bipartite chain graphs in Section 3.
3. We prove k-ary ST and k-ary STREE are NP-complete on strongly chordal split graphs in Section 4.
4. Finally, in Section 5, we show that the results of Section 4 is used to show that k-ary ST and k-ary STREE are NP-complete on $K_{1,r}$-free split graphs, when $r \geq k+4$ and is polynomial-time solvable on $K_{1,r}$-free split graphs, when $r \leq k+2$.

1.1 Preliminaries

In this paper, we work with simple, connected, undirected, and unweighted graphs. For a graph G, let $V(G)$ denote the vertex set and $E(G)$ denote the edge set. The notation $\{u, v\}$ represents an edge incident on the vertices u and v. The neighborhood of a vertex u of G, $N_G(u)$, is the set of vertices adjacent to u in G. The degree of a vertex u is denoted by $d_G(u) = |N_G(u)|$. The maximum degree of a graph G is denoted by $\Delta(G)$. A vertex u is said to be a pendant if $d_G(u) = 1$. A graph H is called an induced subgraph of G if for all $u, v \in V(H)$, $\{u, v\} \in E(H)$ if and only if $\{u, v\} \in E(G)$. A tree is a connected acyclic graph. A path P is a tree T with $V(T) = \{u_1, u_2, \dots, u_n\}$, $n \geq 1$ and $E(T) = \{\{u_i, u_{i+1}\} | 1 \leq i \leq n-1\}$. A graph G is said to be connected if there exists a path between every pair of vertices. A tree T is said to be a k-ary tree if the degree of each vertex in T is at most $k+1$. A graph is chordal if every cycle of length at least 4 in G has a chord. A bipartite graph is said to be a chordal bipartite if every cycle of length at least 6 in G has a chord. A graph is strongly chordal if it is chordal and every cycle of even length at least 6 has an odd chord. A bipartite graph $G(X, Y)$ is a chain graph if the vertices can be ordered as $X = \{x_1, x_2, \dots, x_{|X|}\}$ and $Y = \{y_1, y_2, \dots, y_{|Y|}\}$ such that $N(x_1) \subseteq N(x_2) \subseteq N(x_3) \subseteq \dots \subseteq N(x_{|X|})$ and $N(y_1) \subseteq N(y_2) \subseteq N(y_3) \subseteq \dots \subseteq N(y_{|Y|})$. A graph G is a split graph if it can be partitioned into clique K and an independent set I. For a vertex $u \in K$ We denote $N_G^I(u) = N_G(u) \cap I$ and $d_G^I(u) = |N_G^I(u)|$. Then $\Delta_G^I = max\{d_G^I(u)\}$. Similarly, for a set $S \subseteq V(G)$, $N_G^I(S) = \bigcup_{x \in S} N_G(v) \cap I$. A split graph is said to be a $K_{1,r}$-free split graph if G does not contain an induced subgraph isomorphic to $K_{1,r}$. A comb is a tree $T = (A, F)$ where $A = \{a_1, a_2, \dots, a_{2p}\}$ and $F = \{\{a_i, a_{p+i}\} | 1 \leq i \leq p\} \cup \{\{a_i, a_{i+1}\} | 1 \leq i < p\}$. The vertex set $\{a_i | 1 \leq i < p\}$ or the path $a_1 a_2 \dots a_p$ is backbone of the comb, and $a_{p+1}, a_{p+2}, \dots, a_{2p}$ are teeth of the comb. In a k-ary tree T, a vertex $v \in V(T)$ has saturated means that $d_T(v) = k + 1$, and a vertex $u \in V(T)$ has space for means that $d_T(v) < k + 1$ and can be adjacent to $(k + 1) - d_T(v)$ additionally. A caterpillar graph is a tree in which the removal of all pendant vertices results in a chordless path. A caterpillar graph with two levels means that to a vertex in a chordless path, a path on two vertices is adjacent to it.

Theorem 1. *The k-ary spanning tree problem is in NP.*

Proof. Our certificate is a set of edges that are part of the k-ary spanning tree. In polynomial time, we can verify that these edges form a connected graph with $n - 1$ edges and that each vertex is hit by only at most $k + 1$ edges.

Therefore, the k-ary spanning tree problem is in NP. □

Theorem 2. *The k-ary Steiner tree problem is in NP.*

Proof. Our certificate is a set of edges that are part of the k-ary Steiner tree. In polynomial time, we can verify that these edges form a tree with vertices spanning $R \cup S$ and that each vertex is hit by only at most $k + 1$ edges.

Therefore, the k-ary Steiner tree problem is in NP. □

Observation 2. *Let v be a pendant vertex, and let u be the neighbor of v. In any spanning tree T, $\{u, v\} \in E(T)$.*

Observation 3. *If G is an yes-instance of k-ary spanning tree, then the number of pendant vertices for all $v \in V(G)$ is at most k.*

2 Chordal bipartite graphs

We prove that the computational complexity of the k-ary spanning tree problem on chordal bipartite graphs is NP-complete. It is known that HPATH on chordal bipartite graphs is NP-complete [6]. The reduction instances generated have exactly two pendant vertices. We consider such an instance of chordal bipartite graphs and prove the following theorem.

Theorem 3. *For chordal bipartite graphs and any $k \geq 2$, the k-ary spanning tree problem is NP-complete.*

Proof. We know from Theorem 1 that the k-ary spanning tree problem is in NP. We present a deterministic polynomial-time reduction that reduces an instance of HPATH on chordal bipartite graph G to the corresponding chordal bipartite graph instance of the k-ary spanning tree problem of H as follows: Let the pendant vertices in G be a, b. We shall now define $V(H) = V(G) \cup \{w, v\} \cup \{u_i \mid 1 \leq i < k, u \in V(G)\} \cup \{y_i, z_i \mid 1 \leq i < k\}$. The edge set $E(H) = E(G) \cup \{\{a, w\}, \{b, v\}\} \cup \{\{u, u_i\}, \{a, y_i\}, \{b, z_i\} \mid u \in V(G), 1 \leq i < k\}$. Observe that w, x, u_i, y_i, and $z_i, 1 \leq i < k$ are pendant vertices in H. Thus, H remains a chordal bipartite graph. We claim that G is a yes-instance of HPATH if and only if H is a yes-instance of the k-ary spanning tree problem.

($\Rightarrow$) Suppose that there exists a Hamiltonian path P in G. Let the path P be $(a, u, v, \ldots, b)$. We construct a k-ary spanning tree as follows: By our construction, we know that a and b are adjacent to k pendant vertices. Further, each vertex in $V(G) \setminus \{a, b\}$ is adjacent to $k - 1$ pendant vertices in H. Recall that a vertex in a k-ary tree can have a degree at most $k + 1$. Thus, we obtain a k-ary spanning tree of H by including the corresponding pendant vertices for each vertex in the path.

($\Leftarrow$) Suppose that there exists a k-ary spanning tree of H. We claim that removing the pendant vertices from H yields a Hamiltonian path P in G. Suppose that P is not a Hamiltonian path in G. Let H' be the graph obtained after removing the pendant vertices. Observe that the degree of each vertex in H' is at most 2. A connected graph that can be constructed from the degree sequence is a path graph. Note that H' has to span the entire $V(G)$. If P is not a spanning path of G, then it is a contradiction that H has a k-ary spanning tree as a subgraph.

Therefore, for chordal bipartite graphs and any $k \geq 2$, the k-ary spanning tree problem is NP-complete. □

Now, we shall show that the computational complexity of k-ary STREE on chordal bipartite graphs is NP-complete in the following theorem. Our reduction is from HPATH on chordal bipartite graphs [6].

Theorem 4. *For chordal bipartite graphs and any $k \geq 2$, the k-ary Steiner tree problem is NP-complete.*

Proof. We know from Theorem 2 that the k-ary Steiner tree problem is in NP. We present a deterministic polynomial-time reduction that reduces an instance of HPATH on chordal bipartite graphs G to the corresponding instance of k-ary STREE (H, R, l). Our construction of H is as the construction of H in Theorem 3. We claim that claim that G is a yes-instance of HPATH if and only if H is a yes-instance of the k-ary Steiner tree problem $(H, R, l = n)$,where $R = \{u_i \mid 1 \leq i < k\} \cup \{y_i, z_i \mid 1 \leq i < k\}$. The proof is similar to the proof of the claim in Theorem 3. □

3 Bipartite chain graphs

Having shown that the k-ary spanning tree problem is NP-complete on chordal bipartite graphs, it is natural to investigate the computational complexity of the k-ary spanning tree problem in the subclass of chordal bipartite graphs. In this section, we consider the class of bipartite chain graphs which is a subclass of chordal bipartite graphs. Interestingly, we show that the binary spanning tree problem is polynomial-time solvable on bipartite chain graphs. Also, we extend our binary spanning tree result to the k-ary spanning tree problem. We shall fix the following notation to present our results. For a chordal bipartite G with bipartition (A, B), let $A = \{x_1, x_2, \ldots, x_m\}$ and $B = \{y_1, y_2, \ldots, y_n\}$. The edge set $E(G) \subseteq \{\{x, y\} \mid x \in A, y \in B\}$. Let $A = A_1 \cup A_2$ and $B = B_1 \cup B_2$, $A_1 = \{x_1, x_2, \ldots, x_i\}$, $A_2 = \{u_{i+1}, u_{i+2}, \ldots, u_m\}$ and $B_1 = \{y_1, y_2, \ldots, y_j\}$, $B_2 = \{v_{j+1}, v_{j+2}, \ldots, v_n\}$, such that (A_1, B_1) is a base biclique. The base biclique (A_1, B_1) is a maximal clique $K_{i,j}$ such that $|i - j|$ is minimum over all such maximal bicliques.

Lemma 1. *Let G be a bipartite chain graph. Then, $\forall u \in A_2$, $N_G(u) \subset B_1$ and $\forall v \in B_2$, $N_G(v) \subset A_1$.*

Proof. On the contrary, there exists $u \in A_2$, $N(u) \not\subset B_1$. Case 1: $N(u)$=B_1. Then, $(A_1 \cup \{u\}, B_1)$ is the base biclique, a contradiction. Case 2: $N(u) \subseteq B_2$. This implies that there exists $v \in B_2$ such that $v \in N(u)$. Since $A_1 \cup B_1$ is a maximal base clique, for any vertex $u \in A_1$, $d_G(u) \geq d(v)$. By the property of the bipartite chain graph, all the vertices in A_1 must be adjacent to v. Then, $(A_1, B_1 \cup \{v\})$ is a base biclique of larger size, a contradiction. Case 3: $N(u) \cap B_1 \neq \emptyset$ and $N(u) \cap B_2 \neq \emptyset$. I.e., $\exists u \in A_2$ such that $y_a, v_b \in N(u)$ and $y_a \in B_1$, $v_b \in B_2$. An argument similar to Case 2 shows that all the vertices in A_1 must be adjacent to v_b. Then, $(A_1, B_1 \cup \{v\})$ is a base biclique, a contradiction. Similarly, $\forall v \in B_2$, $N(v) \subset A_1$, can be proved. □

We shall now introduce *Nested Neighborhood Ordering (NNO)* among the vertices of G. Let G be a bipartite chain graph with (A_1, B_1) being the base biclique. Let $A_2 = (u_1, u_2, ..., u_p)$ and $B_2 = (v_1, v_2, ..., v_q)$ are orderings of vertices. If $d_G(u_1) \leq d_G(u_2) \leq d_G(u_3) \leq \ldots \leq d_G(u_p)$, then $N(u_1) \subseteq N(u_2) \subseteq N(u_3) \subseteq \ldots \subseteq N(u_p)$. Further, if $d_G(v_1) \leq d_G(v_2) \leq d_G(v_3) \leq \ldots \leq d_G(v_q)$, then $N(v_1) \subseteq N(v_2) \subseteq N(v_3) \subseteq \ldots \subseteq N(v_q)$. We refer to the above ordering of vertices as *Nested Neighbourhood Ordering (NNO)* of G. We shall arrange the vertices in A_2 in non-decreasing order of their degrees so that we can work with *NNO* of G.

Theorem 5. *Let $G(A_1 \cup A_2, B_1 \cup B_2)$ be a bipartite chain graph and T be a tree. T is a binary spanning tree of G if and only if G satisfies the following (i) A_2 has an ordering $(u_1, u_2, \ldots, u_p)$, such that $\forall u_g, d(u_g) \geq \lceil \frac{g}{2} \rceil$, $1 \leq g \leq p$ and B_2 has an ordering $(v_1, v_2, \ldots, v_q)$, $\forall v_h, d(v_h) \geq \lceil \frac{h}{2} \rceil$, $1 \leq h \leq q$ and (ii) $|A_1| - \lceil \frac{q}{2} \rceil \leq 3(|B_1| - \lceil \frac{p}{2} \rceil)) - 1$, if $|A_2|$ is even; $|A_1| - \lceil \frac{q}{2} \rceil \leq 2(|B_1| - \lceil \frac{p}{2} \rceil))$, otherwise.*

Proof. Let $A_3 = A_1\backslash\{x_1, x_2, \ldots, x_{\lceil\frac{q}{2}\rceil}\} = \{x_{\lceil\frac{q}{2}\rceil+1}, x_{\lceil\frac{q}{2}\rceil+2}, \ldots, x_{i-1}, x_i\}$. $B_3 = B_1\backslash\{y_1, y_2, \ldots, y_{\lceil\frac{p}{2}\rceil}\} = \{y_{\lceil\frac{p}{2}\rceil+1}, x_{\lceil\frac{p}{2}\rceil+2}, \ldots, y_{j-1}, y_j\}$.
($\Rightarrow$) (i) On the contrary, there exists $u_g \in A_2$ such that u_g is the first vertex in the ordering with $d(u_g) < \lceil\frac{g}{2}\rceil$. That is, for $u_k \in \{u_1, \ldots, u_{g-1}\}$, $d_G(u_k) \geq \lceil\frac{k}{2}\rceil$ and $d(u_g) < \lceil\frac{g}{2}\rceil$. Since G follows *NNO*, $d_G(u_g) = \lceil\frac{g}{2}\rceil - 1$. Let $S = \{u_1, \ldots, u_{g-1}\}$. We know that $|N_G(S)| = \lceil\frac{g-1}{2}\rceil$. Observe that $|S| = 2|N_G(S)|$. From the property of *NNO*, we know that $N(u_1) \subseteq N(u_2) \subseteq \ldots \subseteq N(u_{g-1}) \subseteq N(u_g)$. This implies that $d(u_g) = d(u_{(g-1)})$. Clearly, no binary tree exists in the graph induced on $S \cup \{u_g\} \cup N_G(S)$, a contradiction.
(ii) On the contrary, assume that for $|A_2|$ is even, $|A_3| > 2|B_3|$. Note that in any binary tree T in the graph induced on $A_3 \cup B_3$, the degree of the vertices in T is exactly three. Since for any $w_k \in V(T)$, the degree is three, w_k cannot be extended to get a connected binary tree, a contradiction.
($\Leftarrow$) Consider the following subgraphs T_1, T_2, T_3 of G. We shall now define T_1 as follows. $V(T_1) = \{u_1, \ldots, u_p\} \cup \{y_1, \ldots, y_{\lceil\frac{p}{2}\rceil}\}$, $E(T_1) = \{\{u_i, y_j\} | u_i \in A_2, y_j \in B_1, \{u_i, y_j\} \in E(G)\}$. Since A_2 has an ordering $(u_1, u_2, \ldots, u_p)$ such that $\forall u_g, d(u_g) \geq \lceil\frac{g}{2}\rceil, 1 \leq g \leq p$, we obtain a path $P_1 = (u_1, y_1, u_3, y_2, u_5, \ldots, y_{\lceil\frac{p}{2}\rceil-1}, u_p, y_{\lceil\frac{p}{2}\rceil})$. Observe that $\{u_2, u_4, u_{(\frac{p}{2})}\} \cap V(P_1) = \emptyset$. By the property of *NNO* and the degree constraint, $\{u_{2g}, y_{\lceil\frac{2g}{2}\rceil}\} \in E(T_1), 1 \leq g \leq \frac{p}{2}$. Clearly, the vertex set of $E(P_1) \cup \{\{u_{2g}, y_{\lceil\frac{2g}{2}\rceil}\}\}, 1 \leq g \leq \frac{p}{2}$ induces a binary tree T_1' of T_1. Similarly, $V(T_2) = \{v_1, \ldots, v_q\} \cup \{x_1, \ldots, x_{\lceil\frac{q}{2}\rceil}\}$, $E(T_2) = \{\{v_i, x_j\} | v_i \in B_2, x_j \in A_1, \{v_i, x_j\} \in E(G)\}$. we obtain a path $P_2 = (v_1, x_1, v_3, x_2, v_5, \ldots, x_{\lceil\frac{q}{2}\rceil-1}, v_q, y_{\lceil\frac{q}{2}\rceil})$. The vertex set of $E(P_2) \cup \{\{v_{2h}, y_{\lceil\frac{2h}{2}\rceil}\}\}, 1 \leq h \leq \frac{q}{2}$ induces a binary tree T_2' of T_2. Let $V(T_3) = A_3 \cup B_3$. Since $|A_3| \leq 2|B_3|$, we obtain a path $P_3 = (x_{\lceil\frac{q}{2}\rceil+1}, y_{\lceil\frac{p}{2}\rceil+1}, x_{\lceil\frac{q}{2}\rceil+3}, y_{\lceil\frac{p}{2}\rceil+2}, \ldots, y_{\lceil\frac{p}{2}\rceil+j}, x_{\lceil\frac{q}{2}\rceil+i})$. Observe that $\{x_{\lceil\frac{q}{2}\rceil+2}, x_{\lceil\frac{q}{2}\rceil+4, x_{\lceil\frac{q}{2}\rceil+(i-1)}}\} \cap V(P_3) = \emptyset$. Since $A_3 \cup B_3$ is a complete bipartite graph, we choose the following edges to construct T_3', $\{x_{\lceil\frac{q}{2}\rceil+2k}, y_{\lceil\frac{p}{2}\rceil} + k\} \in E(T_1), \frac{p}{2} \leq k \leq i-1$. Note that $d_G(y_{\frac{p}{2}}) \leq 2$ in T_1', $d_G(x_{\frac{q}{2}}) \leq 2$ in T_2', and $d_G(y_{\frac{p}{2}+i}) \leq 2$ in T_3'. We now finally construct a binary tree T as follows. $V(T) = V(T_1') \cup V(T_2') \cup V(T_3')$, $E(T) = E(T_1') \cup E(T_2') \cup E(T_3') \cup \{\{x_{\frac{q}{2}}, y_{\frac{p}{2}+i}\}, \{x_{\frac{q}{2}+1, y_{\frac{p}{2}}}\}\}$. Note that $V(T) = V(G)$. Thus, we obtain a binary tree T spanning $V(G)$.

$\square$

Theorem 6. *Let G be a bipartite chain graph. The binary spanning tree problem is polynomial-time solvable.*

The proof follows from Theorem 5 as the proof is constructive.

The structure of the graph presented in Theorem 5, can be generalized to obtain the following result.

Theorem 7. *Let G be a bipartite chain graph. The k-ary spanning tree problem is polynomial-time solvable, for each $k \geq 3$.*

Further, from the structure of the graph and because of the following observation, the k-ary Steiner tree problem can be solved by using the k-ary spanning tree algorithm as a black box.

Observation 4. *Let the NNO of vertices in G be $A = (u_1, \ldots, u_p)$, and $B = (v_1, \ldots, v_q)$. If G has a NNO of $V(G)$, then $G - x$, for some $x \in A$ or $x \in B$ has a NNO of $V(G) \setminus \{x\}$.*

Theorem 8. *Let G be a bipartite chain graph. The k-ary Steiner tree problem is polynomial-time solvable, for each $k \geq 2$.*

4 Strongly chordal split graphs

In this section, by giving a polynomial time reduction from HPATH on strongly chordal split graphs to k-ary ST on strongly chordal split graphs, we prove that k-ary ST is NP-complete on strongly chordal split graphs. From the reduction instances of [6], it is known that HPATH with two pendant vertices in strongly chordal split graphs is NP-complete. We shall next present a deterministic polynomial time reduction that reduces an instance of HPATH on strongly chordal split graphs to the corresponding instance of k-ary ST of strongly chordal split graphs. Observe that a strongly chordal split graph has two partitions K and I, where the former is the maximal clique partition, and the latter is the independent set.

Theorem 9. *For strongly chordal split graphs and any $k \geq 2$, the k-ary spanning tree problem is NP-complete.*

Proof. We know from Theorem 1 that the k-ary spanning tree problem is in NP. Now, we map an instance of HPATH on strongly chordal split graphs to corresponding instance of k-ary ST of strongly chordal split graphs as follows; $V(H) = (V' = V(G)) \cup \{w_{ij} \mid v_i \in K, 1 \leq j \leq k-1\} \cup \{x_{ij} \mid u_i \in I, 1 \leq j \leq k-1\} \cup \{y^i_{jl} \mid u_i \in I, 1 \leq j \leq (k-1), 1 \leq l \leq k\} \cup \{d_i \mid 1 \leq i \leq 2\} \cup \{p_i \mid 1 \leq i \leq 2k\}$. Let the pendant vertices in G be $u_1, u_n \in I$. The edge of H is $E(H) = E(G) \cup \{\{w_{ij}, v_i\} \mid 1 \leq i \leq |K|, 1 \leq j \leq k-1\} \cup \{\{x_{ij}, u_i\} \mid 1 \leq i \leq |I|, 1 \leq j \leq k-1\} \cup \{\{x_{il}, v_j\} \mid 1 \leq i \leq |I|, 1 \leq l \leq k-1, 1 \leq j \leq |K|\} \cup \{\{x_{ij}, y^i_{jl}\}\} \mid 1 \leq i \leq |I|, 1 \leq j \leq k-1, 1 \leq l \leq k\} \cup \{\{u_1, d_1\}, \{u_n, d_2\}\} \cup \{\{d_1, p_i\} \mid 1 \leq i \leq k\} \cup \{\{d_2, p_i\} \mid k+1 \leq i \leq 2k\}$. This completes the construction of H.

Observe that each vertex in $V(H) \backslash V(G)$ is either a clique vertex or a pendant I vertex in H. Thus, H remains a strongly chordal split graph. We claim that G has a Hamiltonian path if and only if H has a k-ary spanning tree.

($\Rightarrow$) Let P be a Hamiltonian path of G. Observe that any Hamiltonian path must start and end at u_1, u_n. Let a Hamiltonian path in G be $P = (u_1, v_1, \ldots, v_m, V_n)$, where $|K| = m$, $|I| = n$. Now we shall construct a spanning tree T as follows: for each $u_i \in I$ and $u_i \in P$, $\{u_i, x_{ij}\} \in E(T), 1 \leq i \leq |I|, 1 \leq j \leq k-1$. For each $v_i \in K$ and $v_i \in K$, $\{v_i, w_{ij}\} \in E(T)$, where $1 \leq i \leq |K|$, and $1 \leq j \leq k-1$. Further, for each x_{ij}, $\{x_{ij}, y^i_{jl}\} \in E(T)$, where $1 \leq i \leq |I|, 1 \leq j \leq k-1$, and $1 \leq l \leq k$. Note that T is a spanning tree of H, and is a caterpillar with two levels and a backbone being P. Furthermore, the degree of each vertex in T is at most $k+1$. Thus, T is a k-ary spanning tree.

($\Leftarrow$) Let T be a k-ary spanning tree of H. We claim that T is a caterpillar with two levels. On the contrary, suppose that T is not a caterpillar with two levels. Observe that by Observation 2, each vertex in $V(H) \cap V'$ clique partition of H must be $k-1$ pendant vertices, and each vertex $V(H) \setminus V'$ in clique partition of H must be adjacent k pendant vertices.

A vertex $V(H) \setminus V'$ in clique partition must be adjacent to at most one vertex in V'. On the contrary, if a vertex $v \in V(H) \setminus V'$ in clique partition is adjacent to two vertices in V', then v has a degree more than $k+1$ in T, and contradicts the fact that T is a k-ary spanning tree. Thus, a vertex $V(H) \setminus V'$ in clique partition must be adjacent to at most one vertex in V'.

Further, each vertex $v \in V' \cap K$ must be adjacent to exactly two vertices in V'. On the contrary, if a vertex $v \in V' \cap K$ is adjacent to more than two vertices in $V' \cap I$, then v has degree at least $k+2$ in T, contradicts the fact that T is a k-ary spanning tree of H. Thus, a vertex $v \in V' \cap K$ must be adjacent to exactly two vertices in $V' \cap I$.

Furthermore, we claim that the backbone path of T is a Hamiltonian path of G. On the contrary, suppose that the backbone path of T is a Hamiltonian path of G. Note that $v \in V',\ v \in K$, then v has space for at most two vertices, and $u \in V(H) \setminus V',\ u \in K$ has space for at most one vertex. Suppose that $v \in V',\ v \in K$ and $u \in V(H) \setminus V',\ u \in K$ are adjacent, then u is saturated, and v has space for at most one vertex. Since $|K \cap V'| = |I \cap V'| - 1$, v has to connect two vertices in $I \cap V'$ in order to ensure connectivity. Hence, $v \in V',\ v \in K$ and $u \in V(H) \setminus V',\ u \in K$ cannot be adjacent.

Observe that $K \cap V'$ and $I \cap V'$ must form a backbone path, and T is a k-ary spanning tree with two levels. Thus, $K \cap V'$ and $I \cap V'$ backbone path of T is also a Hamiltonian path of G.

Therefore, k-ary ST on strongly chordal split graphs is NP-complete. □

Now, we shall show that the computational complexity of k-ary STREE on strongly chordal split graphs is NP-complete in the following theorem. Our reduction is from HPATH on strongly chordal split graphs [6].

Theorem 10. *For strongly chordal split graphs and any $k \geq 2$, the k-ary Steiner tree problem is NP-complete.*

Proof. We know from Theorem 2 that the k-ary Steiner tree problem is in NP. We present a deterministic polynomial-time reduction that reduces an instance of HPATH on strongly chordal split graphs G to the corresponding instance of k-ary STREE (H, R, l). Our construction of H is as the construction of H in Theorem 9. We claim that claim that G is a yes-instance of HPATH if and only if H is a yes-instance of the k-ary Steiner tree problem $(H, R, l = n)$,where $R = \{u_i \mid 1 \leq i < k\} \cup \{y_i, z_i \mid 1 \leq i < k\}$. The proof is similar to the proof of the claim in Theorem 9. □

5 $K_{1,r}$-free split graphs

In this section, we prove that for any $k \geq 2$, the k-ary spanning tree problem and the k-ary Steiner tree problem remains NP-complete on $K_{1,r}$-free split graphs, when $r \geq k+4$ and is polynomial-time solvable on $K_{1,r}$-free split graphs, when $r \leq k+2$.

To show the computational complexity of the problems under consideration, we need a specific instance of $K_{1,5}$-free split graphs with $|K| = |I| - 1$ and two pendant vertices in I. Akiyama et al. [1] proved the NP-completeness of the Hamiltonian cycle in planar bipartite graphs with a maximum degree of 3. It is easy to see that if the bipartite graph has different-sized partitions for the vertex set, then it is a NO instance for the Hamiltonian cycle problem. It follows that the Hamiltonian cycle problem in planar bipartite graphs with maximum degree 3 and equal-sized vertex partitions is NP-hard. In [8], a reduction is given from the Hamiltonian cycle problem in a planar bipartite graph $G(A, B)$ with maximum degree 3 and $|A| = |B|$ to the Hamiltonian cycle problem in $K_{1,5}$-free split graph with $|K| = |I|$. Further, in [8] the Hamiltonian cycle problem in $K_{1,5}$-free split graph with $|K| = |I|$ is used to show that HPATH on $K_{1,5}$-free split graphs is NP-complete. We observe that for a given instance of $K_{1,5}$-free split graph $G(K, I)$ with $|K| = |I|$, m instances of $K_{1,5}$-free split graph are created to show the NP-completeness of HPATH, where $m = |\{\{x, y\} \mid x \in K, y \in I\}|$. In each instance, three vertices are added, one to K and the other two vertices to I. This shows that for $K_{1,5}$-free split graphs with $|K| = |I| - 1$ with two pendant vertices, HPATH is NP-complete. We use these special instances of $K_{1,5}$-free split graphs to show that the binary spanning tree problem is NP-complete on $K_{1,6}$-free split graphs.

Theorem 11. *Let $k \geq 2$, and let G be a $K_{1,r}$-free split graphs, where $r \geq k+4$. The k-ary spanning tree problem is NP-complete on $K_{1,r}$-free split graphs, where $r \geq k+4$.*

Proof. We know from Theorem 1 that the k-ary spanning tree problem is in NP. We present a deterministic polynomial-time reduction that reduces an instance of HPATH on $K_{1,5}$-free split graphs G to the corresponding instance of k-ary ST of H. Our construction of H is as the construction of H in Theorem 9. We claim that G is a yes-instance of HPATH if and only if H is a yes-instance of the k-ary Spanning tree problem. The proof is similar to the proof of the claim in Theorem 9. □

We shall next present the polynomial-time algorithm for the binary spanning tree problem on $K_{1,r}$-free split graphs, where $r \leq 4$. We further generalize it for k-ary sapnning tree for any $k \geq 2$, on $K_{1,r}$-free split graphs, where $r \leq k+2$.

Lemma 2. *For $K_{1,r}$-free split graphs, $1 \leq r \leq 2$, the binary spanning tree problem is polynomial-time solvable.*

Proof. We know that $K_{1,1}$-free split graphs are singleton graphs. Similarly $K_{1,2}$-free split graphs are complete graphs. Since the graphs are trivial, we obtain a binary tree T, spanning $V(G)$ in polynomial time.

Lemma 3. *[8] Let G be a split graph. G is $K_{1,3}$-free split graph if and only if one of the following conditions holds.*

(i) $\Delta_G^I \leq 1$, and (ii) If there exists a vertex $u \in K$ such that $d_G^I(u) = 2$, then for all $v \in K$, $N_G^I(u) \cap N_G^I(v) \neq \emptyset$

Observation 5. *Let G be a $K_{1,3}$-free split graph. Then for all $u, v \in K$, $N_G^I(u) \cap N_G^I(v) \leq 2$.*

Observation 6. *For $K_{1,3}$-free split graphs, $\Delta_G^I \leq 2$*

Lemma 4. *[8] Let G be a $K_{1,3}$-free split graph. If $\Delta_G^I = 2$ then $|I| \leq 3$.*

Using the above structural properties, for $K_{1,3}$-free split graphs we solve the binary spanning tree problem in polynomial time.

Theorem 12. *For $K_{1,3}$-free split graphs with $n \geq 4$, the binary spanning tree problem is polynomial-time solvable.*

Due to space constraints, we present the proof of Theorem 12 in Appendix.

Theorem 13. *For $K_{1,4}$-free split graphs, the binary spanning tree problem is polynomial time solvable.*

Proof. From the structural properties of $K_{1,4}$-free split graphs [8], we know that $\Delta_G^I \leq 3$. Since $\Delta_G^I \leq 3$, we have the following three cases; (1) $\Delta_G^I \leq 1$, (2) $\Delta_G^I = 2$ and (3) $\Delta_G^I = 3$. Case 1: $\Delta_G^I \leq 1$. To construct a binary tree T spanning $V(G)$, we first obtain a path P on $|K|$ vertices. Let the path be $P = (x_1, x_2, \ldots, x_{|K|})$. For each vertex $v \in I$, we choose any one of its neighbors from P and add the corresponding edge to obtain T. Observe that T is a binary tree spanning $V(G)$, in particular, T is a comb. Case 2: $\Delta_G^I = 2$. Let $S = \{x_i | x_i \in K, d^I(x_i) = 2\}$. Let H be the graph induced on $V(G) \setminus N^I[S]$. Observe that $\Delta_H^I \leq 1$. We obtain a binary tree T' spanning $V(H)$ similar to Case 1. Since $d^I(x_i) = 2$, we obtain a binary tree in $V(S)$. Note that the obtained tree T' is a comb. Since $d^I(x_i) = 2$, we obtain a binary tree in $V(S)$. Since the degree of the end vertices of the backbone is two, We extend T' by including the removed vertices to the end vertices of the comb. Case 3: $\Delta_G^I = 3$. Let $S = \{x_i | x_i \in K, d^I(x_i) = 3\}$. Let H be the graph induced on $V(G) \setminus N^I[S]$. An argument similar to Case 2 gives a binary tree T spanning $V(G)$ □

Theorem 14. *Let $k > 2$, and let G be a $K_{1,r}$-free split graphs, where $r \leq k+2$. The k-ary spanning tree problem is polynomial-time solvable.*

Proof. We know that from the definition of $K_{1,r+1}$-free split graphs, for every $x \in K$, $d_G^I(x) \leq r$. This leads to the following cases; Case 1: For all $x \in K$, $d_G^I(x) \leq r-1$. We construct a r-ary tree T, spanning $V(G)$ as follows. First, we obtain a path P on $|K|$ vertices. Consider a tree T, $V(T) = V(P) \cup I$ and $E(T) = E(P) \cup \{\{x, y\} | x \in K, y \in I\}$. By the definition of r-ary tree, we know that the degree of each vertex in a r-ary tree is at most $r+1$. Since for all $x \in K$, $d_G^I(x) \leq r-1$, the graph induced on T is a r-ary tree, spanning $V(G)$. (ii) There exist set of vertices $x_i \in K$ such that $d_G^I(x) = r$. Let H be the graph induced on $V(G) \setminus N_G[x_i]$. Similar to the idea followed in Theorem 13, to find the binary spanning tree in $K_{1,4}$-free split graphs can be adopted to obtain a tree T', spanning $V(H)$. Observe that, T' is a r-ary tree, spanning $V(H)$. Note that T' has a path P on $|K|$ vertices. Let x_i, be the end vertex of the path. We extend T' by adding the following edges $\{x_i, x\} \cup \{\{x, y\} | y \in N^I(x) \cap I\}$ to T'. Note that $d_G(x) \leq r + 1$ in T'. Thus we obtain a r-ary tree spanning $V(G)$. $\square$

Similar to the k-ary Steiner tree problem result of strongly chordal split graphs, the k-ary Spanning tree problem on split graphs result can be generalized to the k-ary Steiner tree problem on split graphs.

Conclusions and Directions for further research. We have analyzed the computational complexity of the k-ary spanning tree problem and the k-ary Steiner tree problem, for each $k \geq 2$ on strongly chordal graphs, split graphs, chordal bipartite graphs, and bipartite chain graphs.

Interestingly, we can observe the computational complexity of these problems cannot be inferred from the complexity of the Hamiltonian path problem or the Steiner tree problem. Especially for strongly chordal graphs, the Steiner problem is polynomial-time solvable, whereas the computational complexity of the k-ary Steiner tree problem, for each $k \geq 2$, is NP-complete on strongly chordal graphs. This tells us that it is necessary to analyze the computational complexity of the k-ary spanning tree problem and the k-ary Steiner tree problem, for each $k \geq 2$.

It is interesting to look at the computational complexity of the k-ary spanning tree problem and the k-ary Steiner tree problem, for each $k \geq 2$ on special graph classes such as interval graphs, and convex bipartite graphs.

References

1. Takanori Akiyama, Takao Nishizeki, and Nobuji Saito. NP-completeness of the Hamiltonian cycle problem for bipartite graphs. *Journal of Information processing*, 3(2):73–76, 1980.
2. Alan A Bertossi and Maurizio A Bonuccelli. Hamiltonian circuits in interval graph generalizations. *Information Processing Letters*, 23(4):195–200, 1986.
3. Andreas Brandstädt and Dieter Kratsch. *On the restriction of some NP-complete graph problems to permutation graphs*. Springer, 1985.
4. Alessandro D'Atri and Marina Moscarini. Distance-hereditary graphs, steiner trees, and connected domination. *SIAM Journal on Computing*, 17(3):521–538, 1988.
5. Ruo-Wei Hung and Maw-Shang Chang. Linear-time algorithms for the Hamiltonian problems on distance-hereditary graphs. *Theoretical Computer Science*, 341(1-3):411–440, 2005.
6. Haiko Müller. Hamiltonian circuits in chordal bipartite graphs. *Discrete Mathematics*, 156(1-3):291–298, 1996.
7. Haiko Müller and Andreas Brandstädt. The NP-completeness of steiner tree and dominating set for chordal bipartite graphs. *Theoretical Computer Science*, 53(2):257–265, 1987.
8. P Renjith and N Sadagopan. Hamiltonian Path in $K_{1,r}$-free Split Graphs-A Dichotomy. In *Conference on Algorithms and Discrete Applied Mathematics*, pages 30–44. Springer, 2018.
9. Kevin White, Martin Farber, and William Pulleyblank. Steiner trees, connected domination and strongly chordal graphs. *Networks*, 15(1):109–124, 1985.

6 Appendix

The proof of Theorem 12.

Proof. We know from Lemma 3 that for $K_{1,3}$-free split graphs, $\Delta_G^I \leq 2$. Since $\Delta_G^I \leq 2$, we have the following three cases (1) $\Delta_G^I = 0$, (2) $\Delta_G^I = 1$ and (3) $\Delta_G^I = 2$. Case 1: $\Delta_G^I = 0$. This implies that $I = \emptyset$ in G. Since G is clique, we obtain a path P on $|K|$ vertices, which is a binary tree T spanning $V(G)$. Case 2: $\Delta_G^I = 1$. Since $\Delta_G^I = 1$, G is $K_{1,3}$-free and $n \geq 4$, the vertices of I cannot have a common neighbor in K. This implies that each vertex in I must have a unique neighbor. This shows that $|K| \geq |I|$. The construction of binary tree T spanning $V(G)$ as follows: (i) Obtain a path P on $|K|$ vertices. Let the path be $P = (x_1, x_2, \ldots, x_{|K|})$. (ii) Now consider $V(T) = V(P) \cup I$ and $E(T) = E(P) \cup \{\{x_i, y_j\} | \{x_i, y_j\} \in E(G), x_i \in K, y_j \in I\}$. The graph induced on T is a binary tree, spanning $V(G)$. Case 3: $\Delta_G^I = 2$. Since $\Delta_G^I = 2$, it is known from Lemma 4 that $|I| \leq 3$. Case 3.1: $|I| \leq 1$. We first obtain a path P on $|K|$ vertices. When $|I| = 0$, the graph induced on $V(P)$ is a binary tree spanning $V(G)$. In case of $|I| = 1$, the graph induced on T is a binary tree spanning $V(G)$, $V(T) = V(P) \cup I$, $E(T) = E(P) \cup \{\{y, x\} | y \in I, x \in K \cap N_G(y)\}$. Case 3.2: $|I| = 2$. Let $I = \{y_1, y_2\}$. Case 3.2.1: $N_G(y_1) \cap N_G(y_2) = \emptyset$. Without loss of generality, assume that $N_G(y_1) = \{x_1\}$ and $N_G(y_2) = \{x_2\}$. Obtain a path P on $|K|$ vertices. We define the tree T as follows: $V(T) = V(P) \cup I$ and $E(T) = E(P) \cup \{\{x_1, y_1\}, \{x_2, y_2\}\}$. Clearly, the graph induced on T is a binary tree, spanning $V(G)$. Case 3.2.2: $N_G(y_1) \cap N_G(y_2) \neq \emptyset$. Without loss of generality, assume that $N_G(y_1) \cap N_G(y_2) = \{x_1\}$. Obtain a path P on K vertices starting at the vertex x_1. Consider $V(T) = V(P) \cup I$ and $E(T) = E(P) \cup \{\{x_1, y_1\}, \{x_1, y_2\}\}$. It is easy to see that the degree of x_1 is three in T. Thus T is a binary tree, spanning $V(G)$. Case 3.3: $|I| = 3$. Let $I = \{y_1, y_2, y_3\}$. Since G is $K_{1,3}$-free split, for all $x \in K$, $|N_G^I(x)| \leq 2$. This implies that $N_G(y_1) \cap N_G(y_2) \cap N_G(y_3) = \emptyset$. Case 3.3.1: For any $y_i, y_j \in I$, $N_G(y_i) \cap N_G(y_j) = \emptyset$. Without loss of generality, assume that $N_G(y_1) \cap N_G(y_2) = \emptyset$. We obtain a tree T' similar to that of Case 3.2.1. Let $N_G(y_3) = x_i$ and $x_i \notin N_G(y_1) \in N_G(y_2)$. We now construct the binary tree T spanning $V(G)$ from T' as follows: $V(T) = V(T') \cup \{y_3\}$ and $E(T) = E(T') \cup \{\{x_i, y_3\} | x_i \in K, y_3 \in I\}$. The graph induced on T is a binary tree, spanning $V(G)$. Case 3.3.2: There exists $y_i, y_j \in I$, $N_G(y_i) \cap N_G(y_j) \neq \emptyset$. We obtain a binary tree T, spanning $V(G)$ similar to Case 3.2.2. It clear from all cases that finding a binary spanning tree in $K_{1,3}$-free split graphs is polynomial-time solvable. □