AUSTRALASIAN JOURNAL OF COMBINATORICS
Volume **75(2)** (2019), Pages 174–189

Counting peaks on graphs

Alexander Diaz-Lopez

Department of Mathematics & Statistics
Villanova University, Villanova, PA 19085, U.S.A.
alexander.diaz-lopez@villanova.edu

Lucas Everham

Department of Mathematics, Florida Gulf Coast University
Fort Myers, FL 33965, U.S.A.
leverham@fgcu.edu

Pamela E. Harris

Department of Mathematics and Statistics
Williams College, Williamstown, MA 01267, U.S.A.
pamela.e.harris@williams.edu

Erik Insko Vincent Marcantonio

Department of Mathematics, Florida Gulf Coast University
Fort Myers, FL 33965, U.S.A.
einsko@fgcu.edu vrmarcantonio7740@eagle.fgcu.edu

Mohamed Omar

Department of Mathematics, Harvey Mudd College
Claremont, CA 91711, U.S.A.
omar@g.hmc.edu

Abstract

Given a graph G with n vertices and a bijective labeling of the vertices using the integers $1, 2, \ldots, n$, we say G has a peak at vertex v if the degree of v is greater than or equal to 2, and if the label on v is larger than the label of all its neighbors. Fix a set $S \subset V(G)$. We want to determine the number of distinct bijective labelings of the vertices of G, such that the vertices in S are precisely the peaks of G. The set S is called the *peak set of the graph* G, and the set of all labelings with peak set S is denoted by $P(S; G)$. This definition generalizes the study of peak sets of

ISSN: 2202-3518 ©The author(s). Released under the CC BY 4.0 International License

permutations, as that work is the special case of G being the path graph on n vertices. In this paper, we present an algorithm for constructing all of the bijective labelings in $P(S;G)$ for any $S \subseteq V(G)$. We also use combinatorial methods to explore peak sets in certain well-studied families of graphs.

1 Introduction

Let $[n] := \{1, 2, \ldots, n\}$ and $\mathfrak{S}_n$ denote the symmetric group on n letters. We let $\pi = \pi_1\pi_2 \cdots \pi_n$ denote the one-line notation for a permutation $\pi \in \mathfrak{S}_n$ and we say that π has a *peak* at index i if $\pi_{i-1} < \pi_i > \pi_{i+1}$. The peak set of a permutation π is defined as the set

$$P(\pi) = \{i \in [n] \mid \pi \text{ has a peak at index } i\}.$$

Given a subset $S \subseteq [n]$ we denote the set of all permutations with peak set S by

$$P(S;n) = \{\pi \in \mathfrak{S}_n \mid P(\pi) = S\}.$$

Peak sets of permutations have been the focus of much research; in particular, these sets are useful in studying peak algebras of symmetric groups [1, 2, 3, 12, 13, 14], and more recently, enumerating the sets $P(S;n)$ for various S has drawn considerable attention [4, 5, 6, 7, 9, 10, 11]. In their celebrated paper, Billey, Burdzy, and Sagan [5] developed a recursive formula (whose terms alternate in sign) for $|P(S;n)|$ and showed that

$$|P(S;n)| = 2^{n-|S|-1} p_S(n) \tag{1}$$

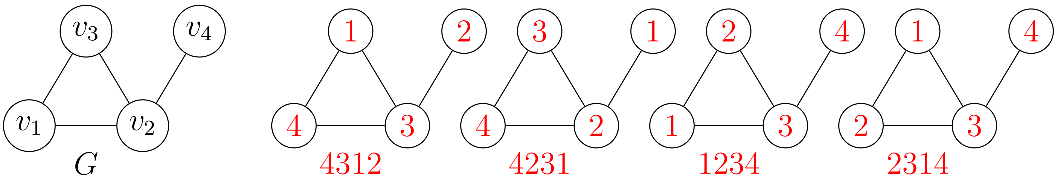
where $p_S(x)$ is a polynomial of degree $\max(S) - 1$ referred to as the *peak polynomial* associated to the set S.

The results in [5] were motivated by a probabilistic meteor mass redistribution model on finite graphs studied by Billey, Burdzy, Pal, and Sagan [4], where it was noted that the concept of a crater is essentially equivalent to that of a peak in a random permutation. The results in [5] also led to subsequent analysis of subsets of Coxeter groups of classical types with given peak sets, including the work of Castro-Velez, Diaz-Lopez, Orellana, Pastrana, and Zevallos [7] and Diaz-Lopez, Harris, Insko, and Perez-Lavin [10]. More recently, Diaz-Lopez, Harris, Insko, and Omar developed a recursive formula for $|P(S;n)|$ that allowed the authors to resolve a conjecture of Billey, Burdzy and Sagan claiming that peak polynomials have nonnegative coefficients when expanded in a particular binomial basis [9]; this newer recursive formula is based on an analysis of the possible positions in which the largest number in the permutation can appear.

In this article, we study a more general version of the original problem of Billey, Burdzy, and Sagan, that of studying peaks on general graphs. We present an algorithm that determines, for a finite graph and a finite set of vertices S deemed to be peaks, the set of labelings where peaks are precisely at vertices in S. The key insight in this process is generalizing the more recent recursive formula for peaks on permutations presented in [9].

To make this precise, let G be a graph with n vertices $v_1, \dots, v_n$. A permutation $\pi = \pi_1 \cdots \pi_n \in \mathfrak{S}_n$ corresponds to a bijective labeling $\ell_\pi : V(G) \to [n]$ by setting π_i to be the label of vertex v_i i.e., $\ell_\pi(v_i) = \pi_i$. Through this correspondence, we interchangeably refer to a labeling and its corresponding permutation. We say that a permutation π has a peak at the vertex v_i of G if $\ell_\pi(v_i) > \ell_\pi(v_j)$ for all vertices v_j adjacent to v_i, and remark that we do not allow peaks at vertices of degree 1 or 0 so that peak sets on paths agree with the existing literature with peak sets on permutations.

Example 1.1. Below is a graph G with vertices v_1, v_2, v_3 and v_4 and four of the 4! different labelings of G. The first two labelings have a peak at v_1, whereas the last two have no peaks.



The G-*peak set* of a permutation π is defined to be the set

$$P_G(\pi) = \{i \in [n] \mid \pi \text{ has a peak at the vertex } v_i\}.$$

Given $S \subseteq V(G) := \{v_1, \dots, v_n\}$, we denote the set of all permutations with G-peak set S by

$$P(S; G) = \{\pi \in \mathfrak{S}_n \mid P_G(\pi) = S\}$$

and say S is a G-admissible set if $P(S; G)$ is nonempty. We classify G-admissible peak sets in Subsection 2.1. Later in Subsection 2.3 we show that the graph G in Example 1.1 has $\{v_1\}$ as an admissible peak set with

$$P(\{v_1\}; G) = \{4321, 4312, 4231, 4132, 4213, 4123, 3124, 3214\}.$$

The paper proceeds as follows. In Section 2, we provide Algorithm 1 for constructing all permutations in $P(S; G)$ for an arbitrary graph G. In Section 3, we explore when combinatorial arguments can be used to find $|P(S; G)|$ directly; in particular we consider graph joins and provide a collection of interesting special cases that show that $|P(S; G)|$ often demonstrates factorial growth, and that the peak polynomials appearing in Equation (1) are rare occurrences.

2 Recursive Construction for Peaks on Graphs

The main goal of this section is to present an algorithm that yields a construction of the set $P(S; G)$ for arbitrary graphs G. Prior to doing so, we classify all admissible peak sets in Subsection 2.1. Next we illustrate our algorithm and notation in

Subsection 2.2 using the example of cycle graphs. Finally, we present the algorithm in Subsection 2.3 and prove it produces $P(S;G)$ in Theorem 2.8.

2.1 Admissible Peak Sets on Graphs

The results in this subsection characterize the G-admissible peak sets of a graph G.

Proposition 2.1. *The set $S = \emptyset$ is a G-admissible peak set if and only if there exists $v \in V(G)$ with $deg(v) = 1$.*

Proof. If G has no vertices of degree 1, then the vertex labeled n is larger than the labels of any of its neighbors. Thus it must be a peak, and $S = \emptyset$ is not an admissible peak set. Conversely, if G has a vertex $v \in V(G)$ of degree one, then one can create a labeling of G with an empty peak set through the following process: (1) starting with v, create a list L of length n by ordering the vertices in $V(G)$ by their distance from v and breaking ties arbitrarily; (2) label v with n and the remaining vertices of L by $n-1, n-2, \ldots, 1$ in descending order. The resulting labeling has no peaks as n appears at the leaf v, and every other vertex in $V(G)$ that is distance k from v has at least one neighbor that is distance $k-1$ from v with a larger label than its own. □

Recall that an independent set is a subset $S \subseteq V(G)$ of vertices in a graph G, no two of which are adjacent. The next result characterizes non-empty admissible peak sets in terms of independent sets.

Proposition 2.2. *A non-empty set $S \subset V(G)$ is G-admissible if and only if S is an independent set containing no degree 1 vertices.*

Proof. If $S \subset V(G)$ is an independent set containing no degree 1 vertices, then one can create a labeling of G with peak set S in the following manner: Label the elements of S with the largest values $n, n-1, \ldots, n-|S|+1$. Order the remaining vertices by their minimum distance from S, breaking ties arbitrarily, and label them with $n-|S|, n-|S|-1, \ldots, 2, 1$ in descending order. The resulting labeling has peaks at every element of S, and any other vertex in $V(G) \setminus S$ that is distance k from S has a neighbor that is distance $k-1$ from S that has a larger label. Hence S is admissible. Conversely, if S is an independent set that contains a vertex of degree 1, then S is not admissible as it is impossible to have a peak at a vertex of degree 1 by definition. □

From these two results we note that the number of non-empty G-admissible sets is equal to the number of independent sets that do not contain leaves, whenever G has at least one leaf, and is one less than the number of independent sets whenever G does not have a leaf.

2.2 Cycles

Let C_n denote a cycle graph on n vertices which we label by $v_1, v_2, \ldots, v_n$. We say that a set $S = \{v_{i_1}, v_{i_2}, \ldots, v_{i_\ell}\} \subseteq V(C_n)$ is *C_n-admissible* if $P(S; C_n) \neq \emptyset$. For each $1 \leq k \leq \ell$ let $\widehat{S}_{i_k} = S \setminus \{v_{i_k}\}$. If S is C_n-admissible, then the label n must be placed at a vertex v_i in S. By removing the vertex v_i and its incident edges from C_n, we obtain $C_n \setminus \{v_i\}$ a path graph on $n-1$ vertices whose peak set is $\widehat{S}_i$. We can now state our first set of results.

Proposition 2.3. *If $S \subset V(C_n)$ is a C_n-admissible set, then*

$$|P(S; C_n)| = \sum_{v_i \in S} |P(\widehat{S}_i\,;\, C_n \setminus \{v_i\})|.$$

Proposition 2.3 follows from Theorem 2.8 in Subsection 2.3 , and hence we omit the details.

Corollary 2.4. *If S is a C_n-admissible set, then* $|P(S; C_n)| = 2^{n-|S|-1} \sum_{v_i \in S} p_{\widehat{S}_i}(n-1)$ *where* $p_{\widehat{S}_i}$ *denotes the peak polynomial of Equation* (1).

Proof. Using Proposition 2.3 and Equation (1), we get

$$\begin{aligned} |P(S; C_n)| &= \sum_{v_i \in S} |P(\widehat{S}_i\,;\, C \setminus \{v_i\})| = \sum_{v_i \in S} |P(\widehat{S}_i\,;\, P_{n-1})| \\ &= \sum_{v_i \in S} 2^{(n-1)-|\widehat{S}_i|-1} p_{\widehat{S}_i}(n-1) = 2^{n-|S|-1} \sum_{v_i \in S} p_{\widehat{S}_i}(n-1). \end{aligned}$$ □

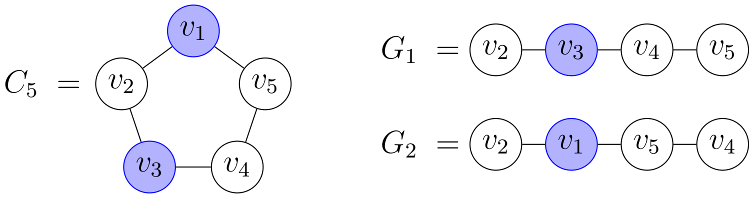


Figure 2.1: Cycle graph on 5 vertices and path graphs G_1 and G_2 obtained from removing vertices v_3 and v_1 from C_5, respectively.

Example 2.5. Consider the graph C_5 in Figure 2.1. If $S = \{v_1, v_3\} \subseteq V(C_5)$ then the sets $\widehat{S}_1 = \{v_3\}$ and $\widehat{S}_3 = \{v_1\}$. One can verify that

$$P(\widehat{S}_1 = \{v_3\}\,;\, G_1) = P(\widehat{S}_3 = \{v_1\}\,;\, G_2) = \{1324, 2314, 1432, 1423, 2431, 3421, 3412\}$$

where G_1 and G_2 are isomorphic to P_4 as shown in Figure 2.1. Proposition 2.3 yields

$$|P(S; C_5)| = |P(\widehat{S}_1\,; G_1)| + |P(\widehat{S}_3\,; G_2)| = 16$$

and one can verify that in fact

$$P(S; C_5) = \begin{Bmatrix} 31542, & 32541, & 41523, & 41532, & 42513, & 42531, & 43512, & 43521, \\ 51324, & 51423, & 51432, & 52314, & 52413, & 52431, & 53412, & 53421 \end{Bmatrix}.$$

For cycle graphs C_n, computing the labelings with peak set S came directly from labeling a vertex v in S with the value n, deleting this vertex labeled n, and then repeating the process on the graph $G \setminus \{v\}$ and its subsequent subgraphs. However, complications may arise when taking this approach for graphs with more complicated neighborhood structure.

For example, consider a graph G with $|V(G)| = n$ and a set $S \subset V(G)$ with $|S| > 1$. We first label a vertex $v \in S$ with n and then remove this vertex. Then, in $G \setminus \{v\}$, the label $n-1$ does not necessarily have to be placed at a vertex in $S \setminus \{v\}$. Instead it could be placed at a neighbor of the vertex v, or at a leaf of G. Another complication with this approach arises when considering the initial vertex on which we place the label n. Because leaves of G cannot be peaks, the label n does not necessarily have to be placed at a vertex in S; instead it can be placed at a leaf of G. Our algorithm correctly manages all of these cases.

2.3 General constructive algorithm for graphs

In this section we describe a recursive algorithm for constructing the set $P(S;G)$ consisting of all labelings of the vertices of a graph G with a given admissible peak set S. We begin by setting the following notation. For any vertex $v \in V(G)$, the *neighborhood* of v, denoted $N_G(v)$ is the set $N_G(v) := \{w \in V(G) : \{v, w\} \text{ is an edge in } G\}$. For any $S \subseteq V(G)$, we let $N_G(S)$ be the neighborhood set of S, namely $N_G(S) = \cup_{v \in S} N_G(v)$. As is standard, we say S is an *independent set* if no vertex in S is in the neighborhood of any other vertex in S, i.e. $S \cap N_G(S) = \emptyset$. Let $V_{<2}(G)$ denote the set of vertices in G of degree less than 2. We now present the algorithm (see next page).

Before verifying Algorithm 1 we present a graphical example that includes every possible choice in line 5 of the algorithm, as well as the algorithm's output.

Example 2.6. Consider the graph G in Figure 2.2, and let $S = \{v_1\}$. We apply Algorithm 1 to G with vertices v_1, v_2, v_3, v_4. Each time we call the function *GraphLabelings*(G, S, L, π), we label a vertex with the largest available number and then remove that vertex from the graph, but for illustrative purposes we color the removed vertices instead of physically removing them from G. Figure 2.2 provides the inputs S, L, π for the first iteration of Algorithm 1, as well as the final output B, which records the labeled graphs. Writing each labeling of G as a permutation, with the label of v_i as the image of i, we obtain

$$P(S;G) = \{4321, 4312, 4231, 4132, 4123, 4213, 3124, 3214\}.$$

The next definition plays an important role in the proof of our main result Theorem 2.8 as it introduces the notation needed to address the possible complications discussed in Subsection 2.2.

Definition 2.7. Let L be a set with $V_{<2}(G) \subseteq L \subseteq V(G)$. Define $P(S, G, L)$ to be the set of labelings of G with peak set S, or peak set S' with $S \subseteq S' \subseteq S \cup (L \setminus N_G(S))$.

```
Algorithm 1: Graph Peak Set Algorithm
   Input  : Graph $G = (V(G), E(G))$, admissible Peak Set $S \subseteq V(G)$, and $L$
            such that $V_{<2}(G) \subseteq L \subseteq V(G)$
   Output: $B$
1  $\pi \leftarrow$ list indexed by $V(G)$ with all entries equal to 0 ;
2  $B \leftarrow$ empty set ;
3  GraphLabelings($G, S, L, \pi$) ;

4  function GraphLabelings($G, S, L, \pi$):
5  for $v \in S \cup (L \setminus N_G(S))$ do
6      $\pi[v] = |V(G)|$ ;
7      if $v \in S$ then
8          $S \leftarrow S \backslash \{v\}$ ;
9      end
10     $L \leftarrow (L \cup N_G(v)) \backslash \{v\}$ ;
11     if $|V(G \backslash \{v\})| > 0$ then
12         GraphLabelings($G \backslash \{v\}, S, L, \pi$) ;
13     end
14     if $|V(G \backslash \{v\})| = 0$ then
15         add $\pi$ to B ;
16         return;
17     end
18 end
19 return $B$ ;
```

If $L = V_{<2}(G)$ then the only potentially admissible peak set S' satisfying $S \subseteq S' \subseteq S \cup (L \setminus N_G(S))$ is S itself because none of the elements of L can be peaks. In this case $P(S, G, L) = P(S; G)$.

Theorem 2.8. *Let S be a G-admissible peak set, L be a set with $V_{<2}(G) \subseteq L \subset V(G)$ and π be a list indexed by $V(G)$ with all entries equal to 0. Then the set $P(S, G, L)$ is equal to the set B which is the output of Algorithm 1 with inputs G, S, and L. Moreover, when $L = V_{<2}(G)$ the set of outputs of GraphLabelings(G, S, L, π) is $P(S; G)$ (the set of labelings of G with peak set S).*

The proof of Theorem 2.8 will result from the following technical lemmas.

Lemma 2.9. *With the assumptions of Theorem 2.8, the set $P(S, G, L)$ is a subset of the set B which is the output of Algorithm 1 with inputs G, S, and L.*

Proof. We prove the result by induction on $|V(G)|$. Let $\mathfrak{L}$ be a labeling in $P(S, G, L)$. First, if $V(G) = \{v\}$ then the only possible peak set is $S = \emptyset$. Since L must contain any vertex of degree less than 2, the only possible L is $\{v\}$. Then $P(S, G, L)$ consists of one labeling, hence $P(S, G, L) = \{[1]\}$ and $\mathfrak{L} = [1]$. When running Algorithm 1 with input G, S and L, in line 3 we call the function *GraphLabelings*(G, S, L, π),

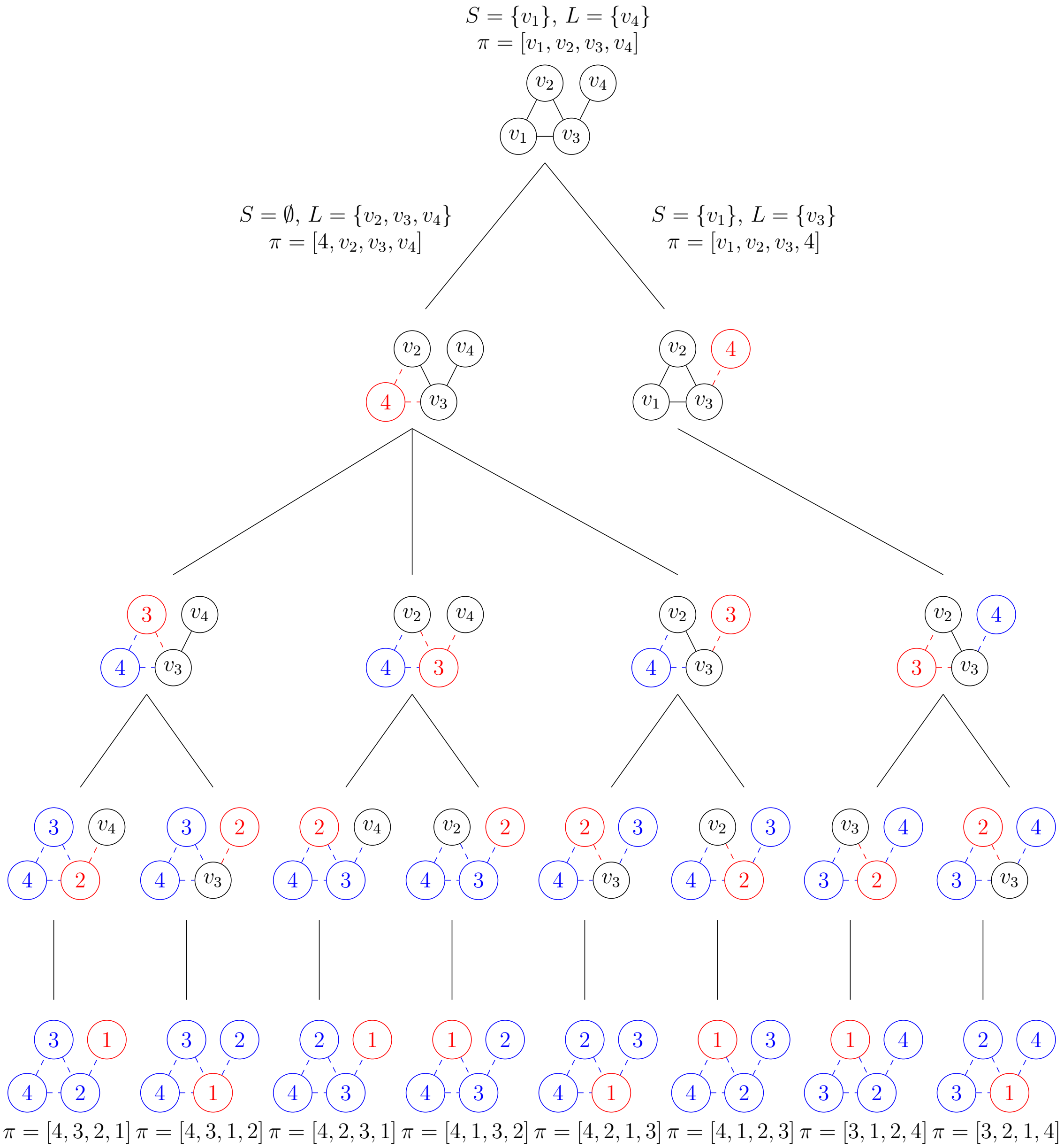

$S = \{v_1\}$, $L = \{v_4\}$
$\pi = [v_1, v_2, v_3, v_4]$
$S = \emptyset$, $L = \{v_2, v_3, v_4\}$
$\pi = [4, v_2, v_3, v_4]$
$S = \{v_1\}$, $L = \{v_3\}$
$\pi = [v_1, v_2, v_3, 4]$
$\pi = [4, 3, 2, 1]$
$\pi = [4, 3, 1, 2]$
$\pi = [4, 2, 3, 1]$
$\pi = [4, 1, 3, 2]$
$\pi = [4, 2, 1, 3]$
$\pi = [4, 1, 2, 3]$
$\pi = [3, 1, 2, 4]$
$\pi = [3, 2, 1, 4]$

Figure 2.2: Application of Algorithm 1 on a small graph G.

where π is defined in line 1. Since G has one vertex v, $GraphLabelings(G, S, L, \pi)$ picks $v \in L \setminus \{N_G(S)\}$ in line 5, labels $\pi[v]$ with 1 in line 6, updates S in lines 7-9, updates L in line 10, skips lines 11-13, and then in line 15, adds $\pi = [1]$ to B. Thus, the set $P(S, G, L)$ is a subset of the set B in this base case.

Now we suppose that the statement is true for all admissible peak sets $S \subseteq V(G')$ and any set L' with $V_{<2}(G') \subseteq L' \subseteq V(G')$ on graphs G' with $1 \leq |V(G')| < n$. Let G be any graph with $|V(G)| = n$, S be an admissible peak set on G, and L be a set with $V_{<2}(G) \subseteq L \subseteq V(G)$. Let $\mathfrak{L}$ be a labeling in $P(S, G, L)$. Recalling Definition 2.7, the peak set $S_{\mathfrak{L}}$ of $\mathfrak{L}$ satisfies

$$S \ \subseteq \ S_{\mathfrak{L}} \ \subseteq \ S \cup (L \setminus N_G(S)). \tag{2}$$

There are two cases to consider depending on where the label n can appear in $\mathfrak{L}$: either n labels a peak or n labels a leaf of G. In other words, either n labels a vertex in S or n labels a vertex in $L \setminus N_G(S)$ with $V_{<2}(G) \subseteq L \subseteq V(G)$. We show that in either case $\mathfrak{L}$ is in B, the output of Algorithm 1 with input G, S, and L.

Case 1: Suppose n labels a vertex v in S. In Algorithm 1, we define π and B in lines 1 and 2, respectively. In line 3 we call the function $GraphLabelings(G, S, L, \pi)$. In line 5 we consider $v \in S$ as the chosen vertex. In line 6, we label v by $\pi[v] = n = |V(G)|$ and then in lines 7-10 we update S to be $S' = S \setminus \{v\}$ and L to be $L' = (L \cup N_G(v)) \setminus \{v\}$. In lines 11-12, we call $GraphLabelings(G \setminus \{v\}, S', L', \pi)$, running Algorithm 1 again with input $G \setminus \{v\}, S'$, and L', except for the caveat that the labeling π has an extra index v labeled with n. Note that by removing v from G we restrict the initial labeling $\mathfrak{L}$ of G to a labeling of $G \setminus \{v\}$, which we denote by $\mathfrak{L}'$. To show $\mathfrak{L}$ in B, it is enough to show that $\mathfrak{L}'$ is in $P(S', G \setminus \{v\}, L')$, which is itself a subset of the output of Algorithm 1 with input $G \setminus \{v\}, S'$, and L' by our induction hypothesis.

The removal of v in G might create peaks at vertices in $N_G(v)$ and we keep all other peaks of $\mathfrak{L}$. Together with (2), this implies that the peak set S'' of $\mathfrak{L}'$ satisfies

$$S' \ \subseteq \ S'' \ \subseteq \ S' \cup (L \setminus N_G(S)) \cup N_G(v). \tag{3}$$

Going back to the definition of $P(S', G \setminus \{v\}, L')$, to show $\mathfrak{L}' \in P(S', G \setminus \{v\}, L')$, we must show that the peak set S'' of $\mathfrak{L}'$ satisfies

$$S' \ \subseteq \ S'' \ \subseteq \ S' \cup \left[\Big((L \cup N_G(v)) \setminus \{v\} \Big) \setminus N_{G\setminus\{v\}}(S') \right]. \tag{4}$$

We will rewrite the rightmost set in (3) to show that the set S'' also satisfies the second inclusion in (4). Note that in (3), in the set $(L \setminus N_G(S)) \cup N_G(v)$, we remove $N_G(v)$ from L and then add it back. Thus, the removal of $N_G(v)$ from L is superfluous. Therefore, $(L \setminus N_G(S)) \cup N_G(v) = (L \setminus N_G(S')) \cup N_G(v)$. Since $v \in S$, it is a peak, and hence it is not adjacent to any other peak. Consequently, $v \notin N_G(S)$, so $(L \setminus N_G(S')) \cup N_G(v) = (L \setminus N_{G\setminus\{v\}}(S')) \cup N_G(v)$. Thus (3) can be rewritten as

$$S' \ \subseteq \ S'' \ \subseteq \ S' \cup (L \setminus N_{G\setminus\{v\}}(S')) \cup N_G(v). \tag{5}$$

Since v is a peak, then v is not a leaf of G, thus (4) can be rewritten as

$$S' \subseteq S'' \subseteq S' \cup \left[\Big(L \cup N_G(v)\Big) \setminus N_{G\setminus\{v\}}(S')\right]. \tag{6}$$

Note that the only elements in the set to the right of the second containment in (5) that are not in the set to the right of the second containment in (6) are the elements in $N_G(v) \cap N_{G\setminus\{v\}}(S')$. Since these elements are in $N_{G\setminus\{v\}}(S')$, they are connected to peaks, hence cannot be peaks themselves. Thus, they cannot be in S'', which proves the second containment in (6).

Case 2: Suppose n labels a vertex v in $L \setminus N_G(S)$. In Algorithm 1, we define π and B in lines 1 and 2, respectively. In line 3 we call the function $GraphLabelings(G, S, L, \pi)$. In line 5 we consider $v \in L \setminus N_G(S)$ as the chosen vertex. We label v by $\pi[v] = n = |V(G)|$ in line 6. Then, we skip lines 7-9 and update L to be $L' = (L \cup N_G(v)) \setminus \{v\}$ in line 10. Then, in lines 11-12, we call $GraphLabelings(G \setminus \{v\}, S, L', \pi)$, running Algorithm 1 again with input $G \setminus \{v\}, S$, and L', except for the caveat that the labeling π has an extra index v labeled with n. Similar to the previous case, it is now enough to show that the labeling $\mathfrak{L}'$, obtained by restricting $\mathfrak{L}$ to the vertices in $V(G) \setminus \{v\}$, is an output of Algorithm 1 with input $G \setminus \{v\}, S$, and L'. We do this by showing that $\mathfrak{L}'$ belongs to $P(S, G \setminus \{v\}, L')$, which is itself a subset the output of Algorithm 1 with input $G \setminus \{v\}, S$, and L' by our induction hypothesis.

Note that removing v from G may create peaks in $\mathfrak{L}'$ at vertices in $N_G(v)$ and we keep all other peaks of $\mathfrak{L}$. This together with (2) implies that the peak set S'' of $\mathfrak{L}'$ satisfies

$$S \subseteq S'' \subseteq S \cup (L \setminus N_G(S)) \cup N_G(v). \tag{7}$$

To prove $\mathfrak{L}' \in P(S, G \setminus \{v\}, L')$, we must show

$$S \subseteq S'' \subseteq S \cup \left[\Big((L \cup N_G(v)) \setminus \{v\}\Big) \setminus N_{G\setminus\{v\}}(S)\right]. \tag{8}$$

We will rewrite the rightmost set in (7) to show that the set S'' also satisfies the second inclusion in (8). Since v had label n, it cannot be adjacent to any peak. Thus, $N_G(S) = N_{G\setminus\{v\}}(S)$. We can replace L in (7) with $L \setminus \{v\}$ because v is no longer a vertex in $G \setminus \{v\}$, thus it cannot be a peak in $\mathfrak{L}'$. We can now rewrite (7) as

$$S \subseteq S'' \subseteq S \cup \big[(L \setminus \{v\}) \setminus N_{G\setminus\{v\}}(S)\big] \cup N_G(v). \tag{9}$$

Since $v \notin N_G(v)$ then we can write $(L \cup N_G(v)) \setminus \{v\}$ as $(L \setminus \{v\}) \cup N_G(v)$. Thus (8) can be written as

$$S \subseteq S'' \subseteq S \cup \left[\Big((L \setminus \{v\}) \cup N_G(v)\Big) \setminus N_{G\setminus\{v\}}(S)\right]. \tag{10}$$

Observe that the only elements in the set to the right of the second containment in (9) that are not in the set to the right of the second containment in (10) are the elements in $N_G(v) \cap N_{G\setminus\{v\}}(S)$. Since these elements are in $N_{G\setminus\{v\}}(S)$, they are connected to peaks, hence cannot be peaks themselves. Thus, they cannot be in S'', which proves the containment in (10). □

Lemma 2.10. *With the assumptions of Theorem 2.8, the output B of Algorithm 1 with inputs G, S, and L is a subset of $P(S, G, L)$.*

Proof. We will show by induction on $|V(G)|$ that any output $\mathfrak{L}$ of B is also an element of $P(G, S, L)$. Let $|V(G)| = 1$, then the only possible peak set is $S = \emptyset$. Since L must contain any vertex of degree less than 2, the only possible L is $\{v\}$. Hence $P(S, G, L)$ consists of one labeling $P(S, G, L) = \{[1]\}$. When running Algorithm 1 with input G, S and L, in line 3 we call the function *GraphLabelings*(G, S, L, π), where π is defined in line 1. Since G has one vertex v, *GraphLabelings*(G, S, L, π) picks $v \in L \setminus \{N_G(S)\}$ in line 5, labels v with 1 in line 6, updates S and L in lines 7-10. Lines 11-13 are skipped because $|V(G \setminus \{v\})| = 0$. In line 15 adds $\pi = [1]$ to B. The algorithm is complete and thus the set $B = \{[1]\}$ is a subset of $P(S, G, L)$ in the base case.

Now we suppose that the statement is true for all admissible peak sets $S \subseteq V(G')$ and any set L' with $V_{<2}(G') \subseteq L' \subseteq V(G')$ on graphs G' with $1 \leq |V(G')| < n$. Let G be a graph with $|V(G)| = n$, S be an admissible peak set, and L be a set satisfying $V_{<2}(G) \subseteq L \subseteq V(G)$. redLet $\mathfrak{L}$ be any labeling in B. Thus, $\mathfrak{L}$ is created via Algorithm 1.

Note that in step 5 of Algorithm 1, a certain vertex of $v \in S \cup (L \setminus \{N_G(S)\})$ is chosen. In step 6, the vertex v is labeled by n, i.e., $\mathfrak{L}[v] = n$. In steps 7-10, some elements are added to the sets S and L. Let S', L' denote the resulting sets. In steps 11-12, we run *GraphLabelings*$(G\setminus\{v\}, S', L', \pi)$, running Algorithm 1 again with input $G \setminus \{v\}, S'$, and L', except for the caveat that the labeling π has an extra index v labeled with n. Let $\mathfrak{L}'$ be the labeling we obtain by restricting $\mathfrak{L}$ to $G \setminus \{v\}$. Then by the induction hypothesis, $\mathfrak{L}'$ is an element of $P(S', G\setminus\{v\}, L')$, i.e., $\mathfrak{L}'$ has peak set $S_{\mathfrak{L}'}$ with $S' \subseteq S_{\mathfrak{L}'} \subseteq S' \cup (L' \setminus N_G(S'))$. To show $\mathfrak{L}$ is an element of $P(S, G, L)$, we must show that when inserting v into $\mathfrak{L}'$ with label n, we get a labeling with peak set $S_{\mathfrak{L}}$ satisfying

$$S \subseteq S_{\mathfrak{L}} \subseteq S \cup (L \setminus N_G(S)). \tag{11}$$

There are two cases to consider.

Case 1: If the chosen vertex v is in S, then $S' = S\setminus\{v\}$ and $L' = (L \cup N_G(v)) \setminus \{v\}$. Thus, $\mathfrak{L}'$ has peak set $S_{\mathfrak{L}'}$ with

$$S' \ \subseteq \ S_{\mathfrak{L}'} \ \subseteq \ S' \cup \left[\left((L \cup N_G(v)) \setminus \{v\}\right) \setminus N_G(S')\right]. \tag{12}$$

By inserting the previously removed label n, indexed by v, back into $\mathfrak{L}'$, we now create a peak at v, and remove any previous peak in vertices that are neighbors of v. Using the containment relations in (12), we get that the peak set $S_{\mathfrak{L}}$ of $\mathfrak{L}$ satisfies

$$S \ \subseteq \ S_{\mathfrak{L}} \ \subseteq \ S \cup \left(L \setminus N_G(S')\right).$$

Since v is a peak, we are certain no vertex in $N_G(v)$ is also a peak, thus we can further say $S \subseteq S_{\mathfrak{L}} \subseteq S \cup (L \setminus N_G(S))$, which proves the containment in (11).

Case 2: We now consider when the chosen vertex v is in $L \setminus N_G(S)$. Then $S' = S$ and $L' = (L \cup N_G(v)) \setminus \{v\}$. Thus, $\mathfrak{L}'$ has peak set $S_{\mathfrak{L}'}$ with

$$S \subseteq S_{\mathfrak{L}'} \subseteq S \cup \left[\Big((L \cup N_G(v)) \setminus \{v\}\Big) \setminus N_G(S)\right]. \tag{13}$$

By inserting the previously removed label n, indexed by v, back into $\mathfrak{L}'$, we either create a peak at v (if $deg_G(v) \geq 2$) or do not create a peak at v (if $deg_G(v) < 2$). In either case, we also remove any previous peaks in vertices that are neighbors of v. Using the containment relations in (13), we get that the peak set $S_{\mathfrak{L}}$ of $\mathfrak{L}$ satisfies

$$S \subseteq S_{\mathfrak{L}} \subseteq S \cup (L \backslash N_G(S)),$$

which proves the containment in (11). □

Proof of Theorem 2.8. Lemmas 2.9 and 2.10 prove the set B consisting of the outputs of Algorithm 1 with input G, S, and L is $P(S, G, L)$. Recall that when $L = V_{<2}(G)$, the only potentially admissible peak set S' satisfying $S \subseteq S' \subseteq S \cup (L \setminus N_G(S))$ is S itself because none of the elements of L can be peaks. Hence $P(S, G, L) = P(S; G)$ in this case. This concludes the proof of Theorem 2.8. □

3 Graph Joins

In some cases, rather than using the presented algorithm to construct the set $P(S; G)$, one can use combinatorial arguments to determine $|P(S; G)|$. In this section we consider the join of two graphs in terms of the peak sets of the constituent graphs. We prove three main results related to the peak sets of graph joins: Proposition 3.1 on the joins of two arbitrary graphs, Proposition 3.2 on the join of an arbitrary graph with a complete graph, and Proposition 3.3 on the join of an arbitrary graph with a null graph.

We recall that the *join* of G_1 and G_2, denoted $G_1 \vee G_2$, has vertex set $V(G_1) \cup V(G_2)$ and edge set $E(G_1) \cup E(G_2) \cup \{\{v_1, v_2\} \mid v_1 \in V(G_1),\ v_2 \in V(G_2)\}$. For example, let $\overline{K_n}$ be the null graph given by an independent set on n vertices, and K_n be the complete graph on n vertices. A star graph on n vertices is $K_1 \vee \overline{K_{n-1}}$ and a complete bipartite graph is $K_{m,n} = \overline{K_m} \vee \overline{K_n}$ (see Figure 3.1).

We first show that peak sets in $G_1 \vee G_2$ are completely contained in either $V(G_1)$ or $V(G_2)$.

Proposition 3.1. *Let G_1 and G_2 be graphs with vertex sets V_1 and V_2 respectively, each with at least two vertices. Let S be a nonempty $(G_1 \vee G_2)$-admissible peak set. Then $S \subseteq V_1$ or $S \subseteq V_2$.*

Figure 3.1: A star graph on 5 vertices and the complete bipartite graph $K_{3,2}$.

Proof. In any labeling of $G_1 \vee G_2$, there is some vertex v labeled by the number $N = |V_1 \cup V_2|$. Assume the vertex v is in V_1. Then no vertex of V_2 can be a peak because v is adjacent to every vertex in V_2, so $S \subseteq V_1$. Similarly, if $v \in V_2$, then $S \subseteq V_2$. □

The next two results consider an arbitrary graph G and a peak set S and give explicit formulas for the number of labelings of the join of G with either the null graph or the complete graph.

Proposition 3.2. *Let $S \subseteq V(\overline{K_n})$ be nonempty, G be any graph with $|V(G)| > 1$, and $G' = \overline{K_n} \vee G$. Then the set S is G'-admissible in and*

$$|P(S;G')| = |S|! \cdot |V(G)| \cdot (|V(G')| - |S| - 1)!$$

Proof. Let $m = |V(G')|$. First we claim that the vertices in S must be labeled by the set

$$\mathcal{I} = \{m, m-1, \ldots, m - |S| + 1\}.$$

Otherwise there are two possible cases: (1) Some vertex in $V(G)$ will be labeled by an element in $\mathcal{I}$ while some element of S will not. This contradicts that S is the peak set. (2) A vertex in $\overline{K}_n$ (not in S) will be labeled by an element in $\mathcal{I}$. In this case, that vertex would be a peak contradicting that S is the peak set. We conclude that the vertices in S must be labeled by the elements of $\mathcal{I}$.

There are $|S|!$ ways to assign these labels to vertices in S, and in any of these labelings the label $m - |S|$ must be assigned in $V(G)$, otherwise there will be an additional peak in $V(\overline{K_n})$. There are $|V(G)|$ such vertices to assign $m - |S|$, each of which guarantees that none of the vertices in $V(\overline{K_n}) \backslash S$ is a peak. None of the remaining vertices are peaks, so we are free to assign them the labels $1, 2, \ldots, m - |S| - 2, m - |S| - 1$ in any order. This completes the proof. □

A similar result is obtained when replacing $\overline{K_n}$ by K_n in Proposition 3.2 and, as the proof is analogous, we omit the details.

Proposition 3.3. *Let G be an arbitrary graph and let $G' = K_n \vee G$. If $S \subseteq V(K_n)$, then*

$$|P(S;G')| = \begin{cases} (|V(G')| - 1)! & \textit{if } |S| = 1 \\ 0 & \textit{otherwise.} \end{cases}$$

Many graph families can be constructed as the join of two graphs, one of which is a null or a complete graph. We apply Propositions 3.2 and 3.3 to give closed formulas describing $|P(S;G)|$ for any admissible peak set $S \subseteq V(G)$ when G is a star graph, ternary star graph, complete bipartite graph, Dutch windmill graph, wheel graph, fan graph, and cone graph. We compile these results in Table 1 below.

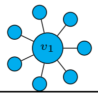
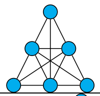
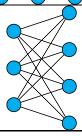
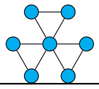
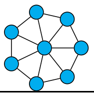
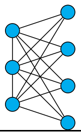
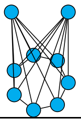
Graph	**Example**	**Results**
Star graph: $\mathcal{S}_n = K_1 \vee \overline{K}_{n-1}$	$\mathcal{S}_8$ 	$\lvert P(S;\mathcal{S}_n)\rvert = \begin{cases} (n-1)!(n-1) & \text{if } S=\emptyset \\ (n-1)! & \text{if } S=\{v_1\} \\ 0 & \text{otherwise.} \end{cases}$
Ternary Stars graph: $k\mathcal{S}_n = K_k \vee \overline{K}_{n-k}$	$3\mathcal{S}_6$ 	$\lvert P(S;k\mathcal{S}_n)\rvert = \begin{cases} (N-1)! & \text{if } S \subset V(K_k) \text{ and } \lvert S\rvert = 1 \\ \lvert S\rvert! \cdot \lvert k\rvert \cdot (N-\lvert S\rvert-1)! & \text{if } S \subset V(\overline{K_{N-k}}) \\ 0 & \text{otherwise.} \end{cases}$
Complete Bipartite: $K_{n,m} = \overline{K}_n \vee \overline{K}_m$	$K_{3,4}$ 	$\lvert P(S;K_{n,m})\rvert = \begin{cases} \lvert S\rvert! \cdot m \cdot (m+n-\lvert S\rvert-1)! & \text{if } S \subset V(\overline{K_n}) \\ \lvert S\rvert! \cdot n \cdot (m+n-\lvert S\rvert-1)! & \text{if } S \subset V(\overline{K_m}) \\ 0 & \text{otherwise.} \end{cases}$
Dutch Windmill graph: $M_k^n = (\cup_{i=1}^n P_2) \vee \overline{K}_1$	M_3^3 	$\left\lvert P(\{v_i\};M_n^k)\right\rvert = (n-1)!$ where v_i is a any noncentral vertex in M_n^k.
Wheel graph: $W_n = C_n \vee \overline{K}_1$	W_8 	$\lvert P(S;W_n)\rvert = n!$ if $S=\{v_1\}$ is the central vertex in W_n.
Fan graph: $F_{n,m} = P_n \vee \overline{K}_m$	$F_{3,4}$ 	$\lvert P(S;F_{n,m})\rvert = \lvert S\rvert! \cdot n \cdot (n+m-\lvert S\rvert-1)!$ if $S \subset V(\overline{K}_m)$.
m-gonal n-cone graph: $C_{m,n} = C_n \vee \overline{K}_m$	$C_{7,2}$ 	$\lvert P(S;C_{m,n})\rvert = \lvert S\rvert! \cdot n \cdot (n+m-\lvert S\rvert-1)!$ if $S \subset V(\overline{K}_m)$.

Table 1: Applications of Propositions 3.2 and 3.3 to certain families of graphs.

4 Future Directions

The following are interesting topics for future consideration.

1. In 2012, Billey, Burdzy, and Sagan conjectured [5, Conjecture 19] and Kasraoui [11, Theorem 1.1] proved that for permutations of n, the most probable peak sets are
   - if $n \equiv 0 \pmod 3$, $\{3,6,9,\ldots\} \cap \{1,2,\ldots,n-1\}$ and $\{4,7,10,\ldots\} \cap \{1,2,\ldots,n-1\}$,
   - if $n \equiv 1 \pmod 3$, $\{3,6,9,\ldots,3s,3s+2,3s+5,\ldots\} \cap \{1,2,\ldots,n-1\}$ with $1 \le s \le \left\lfloor \frac{n}{3} \right\rfloor - 1$ and $|S| = s$,
   - if $n \equiv 2 \pmod 3$, $\{3,6,9,\ldots\} \cap \{1,2,\ldots,n-1\}$.

Table 1 shows for a wheel graph, the empty set is the most probable peak set, and for a complete graph any peak set of size 1 is most probable. For a complete bipartite $K_{n,m} = \overline{K_n} \wedge \overline{K_m}$, with $m > n$, selecting the set $S = V(\overline{K_n})$ to be the largest part, yields the most probable peak set. This leads us to ask, for a graph G, which peak sets are the most probable?

2. Davis, Nelson, Petersen, and Tenner recently introduced the study of pinnacle sets of permutations in which they consider the set S to be the values sitting at the peak positions, rather than the positions themselves [8]. We remark that Algorithm 1 can be modified to create all labelings of a graph with a desired pinnacle set, and pinnacle sets of graphs could be a fruitful direction of future study.

Acknowledgements

We thank Sara Billey for introducing us to this problem. The first author thanks the AMS-Simons Travel Grant. The second, fourth, and fifth authors were supported in part by a Seidler Student/Faculty Undergraduate Scholarly Collaboration Fellowship Program at Florida Gulf Coast University. The third author was partially supported by NSF grant DMS–1620202. We thank two anonymous reviewers for suggestions that substantially improved this paper.

References

[1] M. Aguiar, N. Bergeron and K. Nyman, The peak algebra and the descent algebras of types B and D, *Trans. Amer. Math. Soc.* **356** (7) (2004), 2781–2824.

[2] M. Aguiar, K. Nyman and R. Orellana, New results on the peak algebra, *J. Algebraic Combin.*, **23** (2) (2006), 149–188.

[3] N. Bergeron and C. Hohlweg, Colored peak algebras and Hopf algebras, *J. Algebraic Combin.* **24** (3) (2006) , 299–330.

[4] S. Billey, K. Burdzy, S. Pal and B. Sagan, On meteors, earthworms and WIMPs, *Ann. Appl. Probab.* **25** (4) (2015), 1729–1779.

[5] S. Billey, K. Burdzy and B. Sagan, Permutations with given peak set, *J. Integer Seq.* **16** (6) (2013), Art. 13.6.1, 18 pp.

[6] S. Billey, M. Fahrbach and A. Talmage, Coefficients and Roots of Peak Polynomials, *Exp. Math.* **25** (2) (2016), 165–175.

[7] F. Castro-Velez, A. Diaz-Lopez, R. Orellana, J. Pastrana and R. Zevallos, Number of permutations with same peak set for signed permutations, *J. Comb.* **8** (4) (2017), 631–652.

[8] R. Davis, S. A. Nelson, T. K. Petersen and B. E. Tenner, The pinnacle set of a permutation, *Discrete Math.* **341** (11) (2018), 3249–3270.

[9] A. Diaz-Lopez, P. Harris, E. Insko and M. Omar, A proof of the peak polynomial positivity conjecture, *J. Combin. Theory Ser. A* **149** (2017), 21–29.

[10] A. Diaz-Lopez, P. Harris, E. Insko and D. Perez-Lavin, Peaks sets of classical coxeter groups *Involve* **10** (2) (2017), 263–290.

[11] A. Kasraoui, The most frequent peak set in a random permutation, arXiv:1210.5869, 2012.

[12] K. Nyman, The peak algebra of the symmetric group, *J. Algebraic Combin*, **17** (2003), 309–322.

[13] T. K. Petersen, Enriched P-partitions and peak algebras, *Adv. Math.* **209** (2) (2007), 561–610.

[14] V. Strehl, Enumeration of alternating permutations according to peak sets, *J. Combin. Theory Ser. A* **24** (1978), 238–240.

(Received 24 June 2018; revised 14 June 2019)