AIMS Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 9(6): 14899–14912.
DOI: 10.3934/math.2024721
Received: 31 January 2024
Revised: 09 April 2024
Accepted: 15 April 2024
Published: 24 April 2024

Research article

Enumeration of dissociation sets in grid graphs

Wenke Zhou, Guo Chen, Hongzhi Deng and Jianhua Tu*

School of Mathematics and Statistics, Beijing Technology and Business University, Beijing, 100048, China

* **Correspondence:** Email: tujh81@163.com.

Abstract: A dissociation set of a graph G refers to a set of vertices inducing a subgraph with maximum degree at most 1 and serves as a generalization of two fundamental concepts in graph theory: Independent sets and induced matchings. The enumeration of specific substructures in grid graphs has been a captivating area of research in graph theory. Over the past few decades, the enumeration problems related to various structures in grid graphs such as Hamiltonian cycles, Hamiltonian paths, independent sets, maximal independent sets, and dominating sets have been deeply studied. In this paper, we enumerated dissociation sets in grid graphs using the state matrix recursion algorithm.

Keywords: dissociation sets; enumeration; grid graphs; state matrix recursion algorithm
Mathematics Subject Classification: 05C30, 05C69

1. Introduction

Let G be a simple, undirected graph. An independent set of G is a subset of its vertices inducing a subgraph in which each vertex is an isolated vertex. Moreover, an induced matching in a graph G, denoted as M, is a matching where no two edges in M are joined by edges in graph G. One can identify an induced matching by searching for a subset of vertices that induces a 1-regular subgraph. Both independent sets and induced matchings are fundamental concepts in graph theory and have undergone extensive research. A dissociation set of G is a subset of vertices inducing a subgraph in which each vertex has a degree of at most 1, thus generalizing the concepts of independent sets and induced matchings.

The concept of the dissociation set was first introduced in the 1980s by Yannakakis [17] who demonstrated that determining a maximum dissociation set in bipartite graphs is an NP-hard problem. In contrast, finding a maximum independent set in bipartite graphs can be solved in polynomial time. Over the past few decades, scholars have approached the dissociation set from various perspectives [1, 2, 6, 12, 13, 15–18]. There has been a surge of interest in exploring the problem of identifying the largest number of dissociation sets (or maximal dissociation sets, or maximum dissociation sets) in some classes of graphs [4, 13, 15, 18].

In this study, we focus on the enumeration of dissociation sets in grid graphs. A grid graph with parameters m and n, denoted as $G_{m\times n}$, is composed of vertices representing all points on a two-dimensional coordinate plane which have coordinates (i, j), where i is an integer and ranges from 0 to $m-1$, and j is an integer and ranges from 0 to $n-1$. The edges of the graph connect pairs of vertices (i, j) and (i', j') that satisfy the condition $|i' - i| + |j' - j| = 1$. Alternatively, the grid graph $G_{m\times n}$ can be viewed as the Cartesian product of two paths, one with m vertices and the other with n vertices. To illustrate this concept, Figure 1 presents an example of a grid graph $G_{6\times 6}$ along with a dissociation set in it. We use solid circles to represent the vertices in the dissociation set.

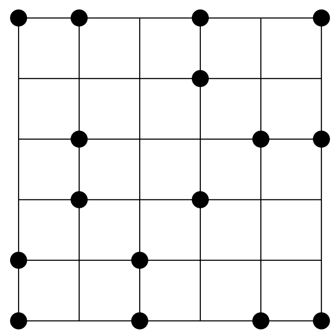

Figure 1. The grid graph $G_{6\times 6}$ and a dissociation set in $G_{6\times 6}$.

We illustrate all the dissociation sets in $G_{m\times n}$ for $1 \le m \le n \le 2$ by giving examples. The empty set is also assumed to be a dissociation set.

When $m = n = 1$, the grid graph $G_{1\times 1}$ is the graph K_1 and contains two distinct dissociation sets.

When $m = 1$ and $n = 2$, Figure 2 illustrates all four dissociation sets in $G_{1\times 2}$.

Figure 2. The four dissociation sets in $G_{1\times 2}$.

When $m = n = 2$, Figure 3 illustrates all eleven dissociation sets in $G_{2\times 2}$.

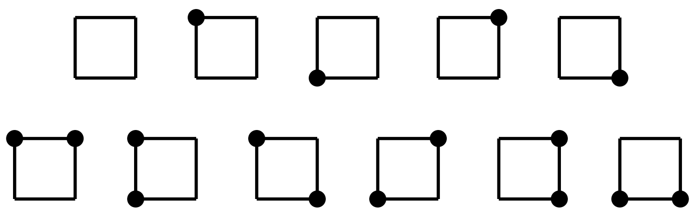

Figure 3. The eleven dissociation sets in $G_{2\times 2}$.

The enumeration problem in grid graphs has its roots in the 1980s, when researchers began exploring the enumeration of Hamiltonian paths and Hamiltonian cycles in grid graphs [5, 14]. Over time, researchers extended this line of inquiry to consider the enumeration of various discrete substructures in grid graphs, including independent sets, maximal independent sets, dominating sets, and more [3, 9–11]. In this study, we target the enumeration of dissociation sets in grid graphs and employ the state matrix recursion algorithm, originally introduced by Oh [11], as a means to tackle this problem.

We derive the dissociation polynomial of a graph G as

$$P_G(z) = \sum_{d=0}^{\phi(G)} y(d) z^d,$$

where $\phi(G)$ is the cardinality of a maximum dissociation set and $y(d)$ is the number of dissociation sets in G containing d vertices. The summation is taken over all dissociation sets of each size in G. The dissociation polynomial of a grid graph $G_{m\times n}$ is simply written as $P_{m\times n}(z)$. Clearly, $P_{m\times n}(1)$ is the total number of all dissociation sets in $G_{m\times n}$.

Let

$$L_m = \begin{pmatrix} 1 & 1 & 0 & 0 \end{pmatrix}^{\otimes m} \text{ and } R_m = \begin{pmatrix} 0 & 1 & 1 & 0 \end{pmatrix}^{\otimes m},$$

where $A^{\otimes m}$ is the m-fold tensor product of a matrix A. Let 0_k be the $4^k \times 4^k$ zero-matrix. Matrices V_m, W_m and X_m are $4^m \times 4^m$ matrices recursively defined by

$$\begin{aligned}
V_{k+1} &= \begin{pmatrix} 0_k & 0_k & 0_k & 0_k \\ V_k + W_k & V_k + W_k & 0_k & 0_k \\ V_k + W_k & V_k + W_k & 0_k & 0_k \\ 0_k & 0_k & 0_k & 0_k \end{pmatrix}, \\
W_{k+1} &= \begin{pmatrix} 0_k & 0_k & z(V_k + X_k) & zV_k \\ 0_k & 0_k & 0_k & 0_k \\ 0_k & 0_k & 0_k & 0_k \\ 0_k & 0_k & zV_k & 0_k \end{pmatrix}, \\
X_{k+1} &= \begin{pmatrix} 0_k & 0_k & zV_k & 0_k \\ 0_k & 0_k & 0_k & 0_k \\ 0_k & 0_k & 0_k & 0_k \\ 0_k & 0_k & 0_k & 0_k \end{pmatrix},
\end{aligned}$$

for $k = 0, \dots, m-1$, starting with

$$V_0 = (1), W_0 = X_0 = (0).$$

Theorem 1.1. *Let R_m^t be the transpose of R_m. Then*

$$P_{m\times n}(z) = L_m \cdot (V_m + W_m)^n \cdot R_m^t.$$

For example, as for $m = n = 2$, we can obtain the expressions of V_2 and W_2 are

$$V_2 = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & z & z & 0 & 0 & z & z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & z & 0 & 0 & 0 & z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & z & z & 0 & 0 & z & z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & z & 0 & 0 & 0 & z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix},$$

$$W_2 = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & z^2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & z & z & 0 & 0 & z & z & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & z & z & 0 & 0 & z & z & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & z & z & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & z & z & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}.$$

The row vectors L_2 and R_2 are

$$L_2 = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$
$$R_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Theorem 1 provides the expression of the dissociation polynomial $P_{2\times2}$ is

$$\begin{aligned} P_{2\times2}(z) &= L_2 \cdot (V_2 + W_2)^2 \cdot R_2^t \\ &= 6z^2 + 4z + 1. \end{aligned}$$

We can obtain that the number of the dissociation sets in $G_{2\times2}$ is equal to $P_{2\times2}(1) = 11$ and the result is in line with prior research.

2. Dissociation polynomials of grid graphs

In this section, we utilize the state matrix recursion algorithm to investigate the enumeration of dissociation sets in grid graphs and prove Theorem 1.1. This algorithm, which has been employed for enumerating independent sets, maximal independent sets, and dominating sets in grid graphs [9–11], consists of the following three stages:

Stage 1. Construct a mosaic system for dissociation sets in $G_{m\times n}$.

Stage 2. Explore the state matrices and recursive matrix-relations.

Stage 3. Derive the dissociation polynomial of $G_{m\times n}$ by analyzing the state matrices and recursive matrix-relations obtained in Stage 2.

2.1. Stage 1

To accurately and effectively represent the states of a quantum knot system, Lomonaco and Kauffman [7, 8] introduced the concept of a mosaic system. In the context of knot mosaic enumeration, Oh [9] formulated a state matrix argument, which later evolved into the state matrix recursion algorithm. This algorithmic advancement allowed Oh to tackle the enumeration of monomer-dimer coverings in grid graphs [9].

Following the terminology and notion in [7, 8], we construct a corresponding mosaic C for F for each dissociation set F in $G_{m\times n}$. In this construction, a tile of C is defined as a square centered at one of the vertices of $G_{m\times n}$. If the vertex at the center of a tile belongs to F, we mark the tile's center with a dot. Every tile's four side edges are labeled with four letters u, v, w and x, according to the following rules.

(1) If a tile contains a dot corresponding to an isolated vertex in the induced subgraph $G_{m\times n}[F]$, all its side edges are labeled with w.
(2) If a tile contains a dot corresponding to a vertex of degree 1 in $G_{m\times n}[F]$, the side edge of it adjoining another tile containing a dot is labeled with x, and the remaining three side edges are labeled with w.
(3) For tiles without dots, the left and right side edges are both labeled with v. As for the top or bottom side edge, it is labeled with u if it adjoins a tile with a dot; otherwise, it is also labeled with v.

Thus, there are in total nine mosaic tiles I_1–I_9 that are shown in Figure 4.

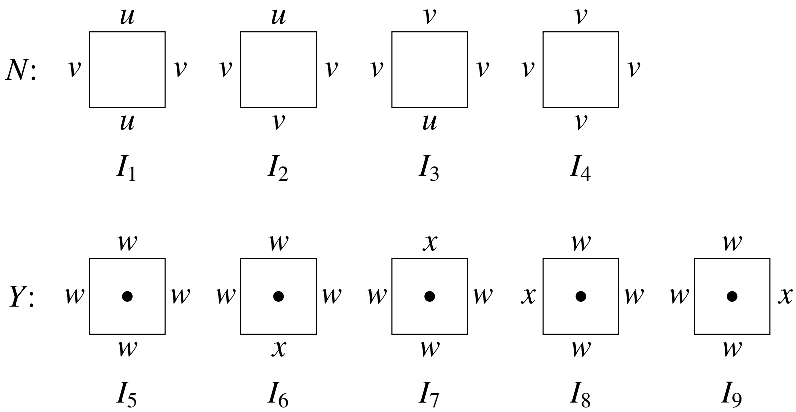

Figure 4. Nine mosaic tiles, and the sets N and Y.

The mosaic corresponding to the dissociation set depicted in Figure 1 is illustrated in Figure 5.

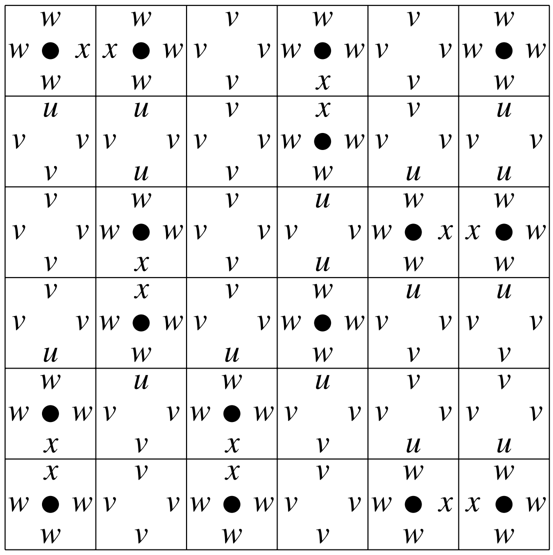

Figure 5. The corresponding mosaic for the dissociation set shown in Figure 1.

Let

$$N = \{I_1, I_2, I_3, I_4\} \text{ and } Y = \{I_5, I_6, I_7, I_8, I_9\}.$$

The sets N and Y are derived by classifying the nine mosaic tiles based on the presence or absence of a dot. Specifically, the side edges of the mosaic tiles belonging to N are exclusively labeled with the letters u and v, whereas the side edges of the mosaic tiles in Y are labeled only with the letters w and x.

In defining an $m \times n$-mosaic, we envision it as an $m \times n$ rectangular array $C = \left(C_{ij}\right)$ of tiles. Each entry C_{ij} represents a mosaic tile positioned in the i-th column from right to left, and the j-th row from top to bottom. Our interest lies only in the mosaics whose tiles match their neighboring tiles correctly to represent dissociation sets. As following are the rules we have created for this purpose.

- **Horizontal adjacency rule:** In a row, the adjacent tiles' abutting edges are labeled with one of the following letter pairs: $v|v$, $v|w$, $x|x$.
- **Vertical adjacency rule:** In a column, the adjacent tiles' abutting edges are labeled with one of the following letter pairs: $u|w$, $v|v$, $x|x$.
- **Boundary rule:** Every boundary edge of a mosaic may be labeled with the letters v or w.

The horizontal adjacency rule and the vertical adjacency rule are shown in Figure 6.

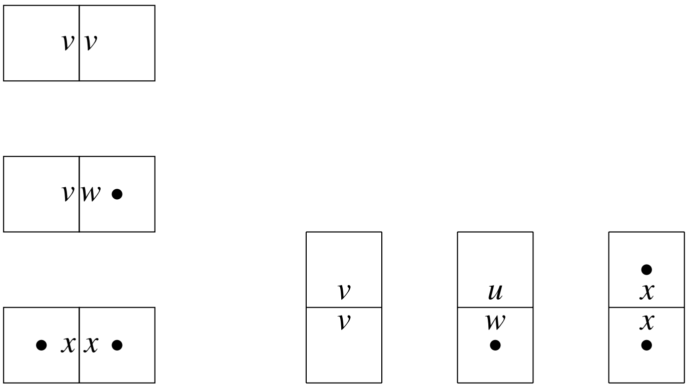

Figure 6. The horizontal adjacency rule and the vertical adjacency rule.

An $m \times n$-mosaic is deemed suitably adjacent when every adjacent pair of tiles adheres to two adjacency rules, furthermore, if it also satisfies the boundary rule, it is called a dissociation set $m \times n$-mosaic.

There exists a one-to-one mapping from dissociation sets in $G_{m\times n}$ to dissociation set $m \times n$-mosaics. It is clear that the number of vertices in a dissociation set equals the number of tiles belonging to the set Y in its corresponding dissociation set $m \times n$-mosaic. Based on the one-to-one mapping, enumerating dissociation sets in $G_{m\times n}$ is equivalent to enumerating dissociation set $m \times n$-mosaics.

2.2. Stage 2

In this subsection, we recall from [10] states and state polynomials. Let $p \leq m$ and $q \leq n$ be two positive integers and C be a suitably adjacent $p \times q$-mosaic. We denote by $d(C)$ the number of tiles of C belonging to the set Y.

We introduce four distinct states for C: The t-state $s_t(C)$, b-state $s_b(C)$, r-state $s_r(C)$, and l-state $s_l(C)$. Each state corresponds to a finite sequence composed of the letters u, v, w, and x. Specifically, the t-state $s_t(C)$ and b-state $s_b(C)$ are sequences of length p derived by reading the labels on the top and bottom boundary edges of C from right to left, respectively. Analogously, the r-state $s_r(C)$ and l-state $s_l(C)$ are sequences of length q derived by reading the labels on the right and left boundary edges of C from top to bottom, respectively. A suitably adjacent mosaic and its four states are shown and described in Figure 7.

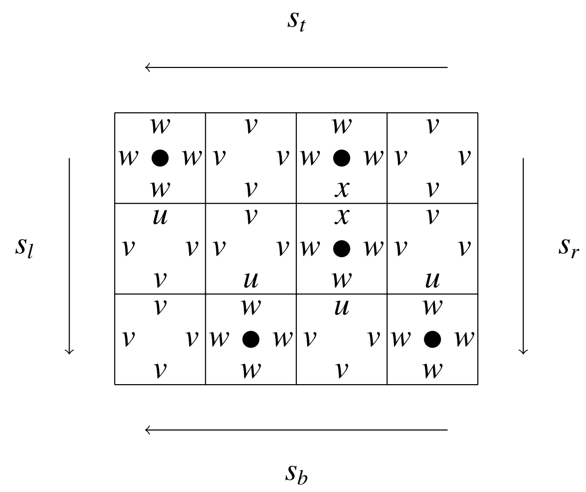

Figure 7. A suitably adjacent mosaic C with $s_t(C) = vwvw$, $s_b(C) = wvwv$, $s_r(C) = vvw$, and $s_l(C) = wvv$.

Given three sequences s_r, s_b and s_t composed of the letters u, v, w and x, we obtain the state polynomial

$$P_{\langle s_r,s_b,s_t\rangle}(z) = \sum_C i(d)z^d,$$

where the summation is taken over all suitably adjacent $p \times q$-mosaics C with the property that $s_r(C) = s_r$, $s_b(C) = s_b$, $s_t(C) = s_t$ and $s_l(C)$ satisfies the boundary rule, and $i(d)$ is the number of suitably adjacent $p \times q$-mosaics C with $d(C) = d$. We use $i_{\langle s_b,s_t\rangle}(d)$ to denote the number of mosaics satisfying b-state is s_b and t-state is s_t. Note that none of $s_r(C)$, $s_b(C)$ and $s_t(C)$ needs to satisfy the boundary rule.

A bar mosaic of length p is a suitably adjacent $p \times 1$-mosaic and has at most 4^p different t- and b-states, which are especially called bar states. The bar states are arranged in the following two orders; (1) The *uvwx*-order (for example, the order of bar states of length $p = 2$ and $p = 3$ is as follows: $p = 2$: *uu*, *uv*, *uw*, *ux*, *vu*, *vv*, *vw*, *vx*, *wu*, *wv*, *ww*, *wx*, *xu*, *xv*, *xw*, *xx*; And $p = 3$: *uuu*, *uuv*, *uuw*, *uux*, *uvu*, *uvv*, *uvw*, *uvx*, *uwu*, *uwv*, *uww*, *uwx*, *uxu*, *uxv*, *uxw*, *uxx*, *vuu*, *vuv*, *vuw*, *vux*, *vvu*, *vvv*, *vvw*, *vvx*, *vwu*, *vwv*, *vww*, *vwx*, *vxu*, *vxv*, *vxw*, *vxx*, *wuu*, *wuv*, *wuw*, *wux*, *wvu*, *wvv*, *wvw*, *wvx*, *wwu*, *wwv*, *www*, *wwx*, *wxu*, *wxv*, *wxw*, *wxx*, *xuu*, *xuv*, *xuw*, *xux*, *xvu*, *xvv*, *xvw*, *xvx*, *xwu*, *xwv*, *xww*, *xwx*, *xxu*, *xxv*, *xxw*, *xxx*.); (2) The *wvux*-order (for example, the order of bar states of length $p = 2$ and $p = 3$ is as follows: $p = 2$: *ww*, *wv*, *wu*, *wx*, *vw*, *vv*, *vu*, *vx*, *uw*, *uv*, *uu*, *ux*, *xw*, *xv*, *xu*, *xx*. $p = 3$: *www*, *wwv*, *wwu*, *wwx*, *wvw*, *wvv*, *wvu*, *wvx*, *wuw*, *wuv*, *wuu*, *wux*, *wxw*, *wxv*, *wxu*, *wxx*, *vww*, *vwv*, *vwu*, *vwx*, *vvw*, *vvv*, *vvu*, *vvx*, *vuw*, *vuv*, *vuu*, *vux*, *vxw*, *vxv*, *vxu*, *vxx*, *uww*, *uwv*, *uwu*, *uwx*, *uvw*, *uvv*, *uvu*, *uvx*, *uuw*, *uuv*, *uuu*, *uux*, *uxw*, *uxv*, *uxu*, *uxx*, *xww*, *xwv*, *xwu*, *xwx*, *xvw*, *xvv*, *xvu*, *xvx*, *xuw*, *xuv*, *xuu*, *xux*, *xxw*, *xxv*, *xxu*, *xxx*). For a positive integer $1 \leq i \leq 4^p$, we denote by ε_i^p and λ_i^p the i-th bar states in the set of states of length p in the *wvux*- and *uvwx*-order, respectively (for example, $\varepsilon_1^2 = ww$, $\lambda_1^2 = uu$).

Bar state matrices V_p, W_p and X_p for the set of suitably adjacent bar mosaics of length p are $4^p \times 4^p$ matrices whose entries v_{ij}, w_{ij}, and x_{ij} are respectively given by

$$v_{ij} = P_{\langle v,\varepsilon_i^p,\lambda_j^p\rangle}(z), \; w_{ij} = P_{\langle w,\varepsilon_i^p,\lambda_j^p\rangle}(z), \text{ and } x_{ij} = P_{\langle x,\varepsilon_i^p,\lambda_j^p\rangle}(z).$$

ε_i^p is the i-th state in *uvwx*-order, and λ_j^p is j-th state in *wvux*-order. So we define ε_i^p and λ_j^p as the row index and column index of $a_{ij}(a = v, w, x)$ respectively.

Lemma 2.1. *The matrices V_p, W_p, and X_p can be recursively derived as follows:*

$$V_{k+1} = \begin{pmatrix} 0_k & 0_k & 0_k & 0_k \\ V_k + W_k & V_k + W_k & 0_k & 0_k \\ V_k + W_k & V_k + W_k & 0_k & 0_k \\ 0_k & 0_k & 0_k & 0_k \end{pmatrix},$$

$$W_{k+1} = \begin{pmatrix} 0_k & 0_k & z(V_k + X_k) & zV_k \\ 0_k & 0_k & 0_k & 0_k \\ 0_k & 0_k & 0_k & 0_k \\ 0_k & 0_k & zV_k & 0_k \end{pmatrix},$$

$$X_{k+1} = \begin{pmatrix} 0_k & 0_k & zV_k & 0_k \\ 0_k & 0_k & 0_k & 0_k \\ 0_k & 0_k & 0_k & 0_k \\ 0_k & 0_k & 0_k & 0_k \end{pmatrix},$$

for $k = 1, \cdots, p-1$, with seed matrices

$$V_1 = \begin{array}{c} \\ w \\ v \\ u \\ x \end{array}\!\!\begin{array}{c} \begin{array}{cccc} u & v & w & x \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}, W_1 = \begin{array}{c} \\ w \\ v \\ u \\ x \end{array}\!\!\begin{array}{c} \begin{array}{cccc} u & v & w & x \end{array} \\ \begin{pmatrix} 0 & 0 & z & z \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & z & 0 \end{pmatrix} \end{array}, X_1 = \begin{array}{c} \\ w \\ v \\ u \\ x \end{array}\!\!\begin{array}{c} \begin{array}{cccc} u & v & w & x \end{array} \\ \begin{pmatrix} 0 & 0 & z & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

Remark. According to the lemma, we can also start with matrices

$$V_0 = (1) \text{ and } W_0 = X_0 = (0).$$

We present a detailed step-by-step process for deriving $v_{i,j}$, $w_{i,j}$ and $x_{i,j}$ while $p = 1$ before proving Lemma 2.

The matrices V_1, W_1 and X_1 are the bar state matrices for three sets of the 1×1 mosaics C whose r-states are v, w and x, respectively. So $v_{i,j} = P_{\langle v,\varepsilon_i^1,\lambda_j^1\rangle}(z)$, $w_{i,j} = P_{\langle w,\varepsilon_i^1,\lambda_j^1\rangle}(z)$ and $x_{i,j} = P_{\langle x,\varepsilon_i^1,\lambda_j^1\rangle}(z)$.

As for the matrix V_1, the row index of $v_{2,1}$ is v and the column index is u, and all possible corresponding mosaic satisfying $s_b(C) = v$ and $s_t(C) = u$ is I_2 shown in Figure 5. Furthermore, $d(I_2) = 0$ and $i_{\langle v,v\rangle}(d(I_2)) = 1$, so $v_{2,1} = P_{\langle v,v,u\rangle}(z) = 1 \cdot z^0 = 1$. Now we write $v_{i,j}$ individually. Initially, we focus on the non-zero entries ($v_{2,1}$ has been been given). The sets of all possible corresponding mosaics for $v_{2,2}$, $v_{3,1}$ and $v_{3,2}$ are $\{I_4\}$, $\{I_1\}$ and $\{I_3\}$ respectively. Furthermore, $d(I_4) = d(I_1) = d(I_3) = 0$ and $i_{\langle v,u\rangle}(d(I_4)) = i_{\langle u,u\rangle}(d(I_1)) = i_{\langle u,v\rangle}(d(I_3)) = 1$, that is $v_{2,2} = v_{3,1} = v_{3,2} = 1$. There is no mosaic C satisfying $s_b(C) = \varepsilon_i^1$ and $s_t(C) = \lambda_j^1$, so the entries $v_{i,j} = P_{\langle v,\varepsilon_i^1,\lambda_j^1\rangle}(z) = 0$ for $i \in \{1,4\}$ or $j \in \{3,4\}$.

Similarly, $w_{1,3} = P_{\langle w,w,w\rangle}(z) = w_{1,4} = P_{\langle w,w,x\rangle}(z) = w_{4,3} = P_{\langle w,x,w\rangle}(z) = z$, because the sets of all possible corresponding mosaics for $w_{1,3}$, $w_{1,4}$ and $w_{4,3}$ are $\{I_5\}$(I_8 doesn't satisfying the boundary rule), $\{I_7\}$, and $\{I_6\}$ respectively. Then $d(I_5) = d(I_7) = d(I_6) = 1$ and $i_{\langle w,w\rangle}(d(I_5)) = i_{\langle w,x\rangle}(d(I_7)) = i_{\langle x,w\rangle}(d(I_6)) = 1$. The elements $x_{i,j}$ of X_1 can also be obtained similarly in this way.

Proof. We prove Lemma 2.1 by induction on k. When $k = 1$, we can obtain the seed matrices by straightforward observations. For example, the (1,3)-entry of W_1 is

$$P_{\langle w,\varepsilon_1^1,\lambda_3^1\rangle}(z) = P_{\langle w,w,w\rangle}(z) = z,$$

because of the following facts: (1) $\varepsilon_1^1 = w$ in the $wvux$-order, (2) $\lambda_3^1 = w$ in the $uvwx$-order, (3) $s_l(C) = w$ because of the boundary rule, (4) there exists a unique mosaic tile I_5 that satisfies $s_l(C) = w$, $s_r(C) = w$, $s_b(C) = w$, and $s_t(C) = w$. Moreover, $d(C) = 1$.

Suppose that V_ℓ, W_ℓ, and X_ℓ have been obtained recursively. Consider the matrix $W_{\ell+1}$, and divide the matrix of size $4^{\ell+1} \times 4^{\ell+1}$ into 16 block submatrices of size $4^\ell \times 4^\ell$. For the (1,3)-submatrix of $W_{\ell+1}$, which is the (1,3)-component lying in the 1st row and 3rd column in the 4×4 array of the 16 blocks, the (i, j)-entry of it is the state polynomial

$$P_{\langle w,\, w\varepsilon_i^\ell,\, w\lambda_j^\ell \rangle}(z),$$

where $w\varepsilon_i^\ell$ is the combination of two states w and ε_i^ℓ, which indicates the rightmost letter of the bottom state is w ($w\lambda_j^\ell$ is similar), in other words, this new state is obtained by reading the letter w before reading the state ε_i^ℓ from right to left. Thus a suitably adjacent $(\ell + 1) \times 1$-mosaic can be obtained by pasting a 1×1-mosaic C' satisfying $s_r(C') = w$, $s_b(C') = w$, and $s_t(C') = w$ to the rightmost of an $\ell \times 1$-mosaic C satisfying $s_b(C) = \varepsilon_i^\ell$ and $s_t(C) = \lambda_j^\ell$. The mosaic tile C' belongs to Y, $d(C') = 1$. The l-state of C' must be w or x, which implies the r-state of the corresponding $\ell \times 1$-mosaic C can only be v or x, as shown in Figure 8.

Figure 8. Expand the bar mosaic $(\ell) \times 1$-mosaic C to the bar mosaic $(\ell + 1) \times 1$-mosaic.

Thus, we have

$$P_{\langle w,\, w\varepsilon_i^\ell,\, w\lambda_j^\ell \rangle}(z) = [\text{the } (i, j)\text{-entry of } (V_\ell + X_\ell)] \cdot z,$$

which implies that the (1,3)-submatrix of $W_{\ell+1}$ is $(V_\ell + X_\ell) \cdot z$. In the same way, we can obtain the other submatrices of $W_{\ell+1}$, $V_{\ell+1}$, and $X_{\ell+1}$.

The proof of Lemma 2.1 is complete. ∎

For the set of suitably adjacent $m \times q$-mosaics, we obtain the state matrix $H_{m\times q}$ as a $4^m \times 4^m$ matrix where the (i, j)-entry is

$$h_{ij} = \sum_{s_r} P_{\langle s_r, \varepsilon_i^m, \lambda_j^m \rangle}(z),$$

which the summation is taken over all r-states s_r of length q satisfying the boundary rule. Moreover, the rows and columns of the state matrix are indexed in the same way as the bar state matrix.

Lemma 2.2.

$$H_{m\times n} = (V_m + W_m)^n.$$

Proof. We prove Lemma 2.2 by induction on n. When $n = 1$, because of the boundary rule, the r-state of the $m \times 1$-mosaics considered can be only v or w. Thus,

$$H_{m\times 1} = V_m + W_m.$$

Suppose that $H_{m\times k} = (V_m + W_m)^k$. For a suitably adjacent $m\times(k+1)$-mosaic $C^{m\times(k+1)}$, by removing the topmost bar mosaic of $C^{m\times(k+1)}$, we can divide $C^{m\times(k+1)}$ into two mosaics: $C^{m\times 1}$ and $C^{m\times k}$. Furthermore, every top boundary side edge of $C^{m\times k}$ and its abutting bottom boundary side edge of $C^{m\times 1}$ must satisfy the vertical adjacency rule. Specifically, when transitioning from $s_t(C^{m\times k})$ to $s_b(C^{m\times 1})$, the letter u is substituted by w, and vice versa, w is replaced by u. Please refer to Figure 9 for further clarification. Thus, there exist some $r \in \{1, \cdots, 4^m\}$ such that $s_b\left(C^{m\times 1}\right) = \varepsilon_r^m$ and $s_t\left(C^{m\times k}\right) = \lambda_r^m$.

Let

$$H_{m\times(k+1)} = \left(h_{ij}\right), H_{m\times k} = \left(h'_{ij}\right), \text{and } H_{m\times 1} = \left(h''_{ij}\right).$$

The entry h_{ij} of $H_{m\times(k+1)}$ is the state polynomial for the set of suitably adjacent $m \times (k+1)$-mosaics C, which can be divided into $C^{m\times 1}$ and $C^{m\times k}$ such that $s_t\left(C^{m\times 1}\right) = s_t(C) = \lambda_j^m$, $s_b\left(C^{m\times k}\right) = s_b(C) = \varepsilon_i^m$, and $s_t\left(C^{m\times k}\right) = \lambda_r^m$ and $s_b\left(C^{m\times 1}\right) = \varepsilon_r^m$ for some $r \in \{1, \cdots, 4^m\}$. So

$$H_{ij} = \sum_{r=1}^{4^m} h'_{ir} \cdot h''_{rj},$$

which implies that

$$H_{m\times(k+1)} = H_{m\times k} \cdot H_{m\times 1} = (V_m + W_m)^{k+1}.$$

The proof of Lemma 2.2 is complete. ∎

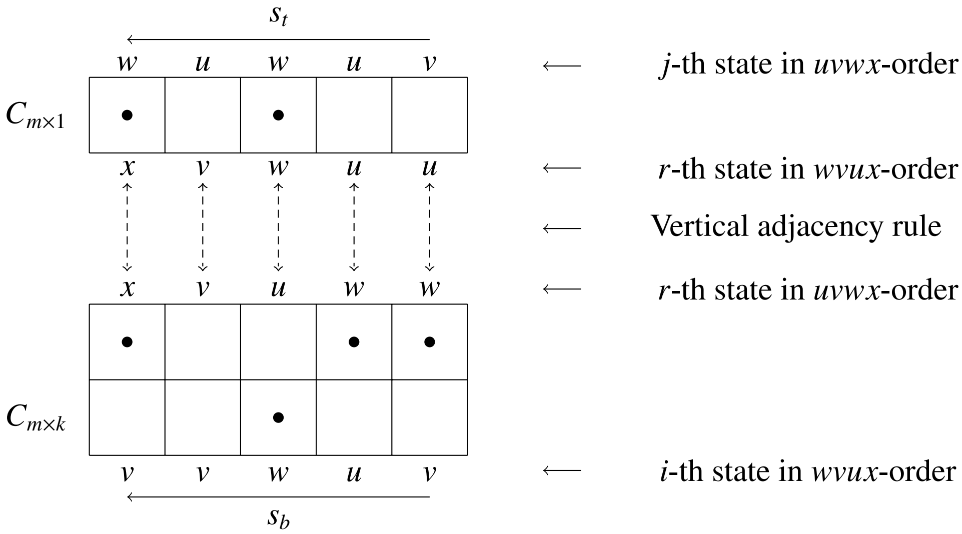

Figure 9. A suitably adjacent $m \times (k+1)$-mosaic $C^{m\times(k+1)}$.

2.3. Stage 3

We are now in a position to prove Theorem 1.1 by analyzing the state matrix $H_{m\times n}$.

Proof of Theorem 1.1. For every suitably adjacent $m\times n$-mosaic C which is considered in the (i, j)-entry of $H_{m\times n}$, the states $s_r(C)$ and $s_l(C)$ satisfy the boundary rule. According to the one-to-one mapping, the dissociation sets in $G_{m\times n}$ correspond to the suitably adjacent $m \times n$-mosaics C whose l-, r-, b-, and t-states satisfy the boundary rule. Hence the dissociation polynomial $P_{m\times n}(z)$ of $G_{m\times n}$ is the sum of all entries h_{ij} whose column index λ_j^m and row index ε_i^m consist of letters v and w. Therefore, $P_{m\times n}(z)$ can be obtained by deleting the entries h_{ij} of $H_{m\times n}$ associated with $s_b(C)$ and $s_t(C)$ consisting of at least one letter of u and x, and then adding up the rest.

Recall that

$$L_m = \begin{pmatrix} 1 & 1 & 0 & 0 \end{pmatrix}^{\otimes m} \text{ and } R_m = \begin{pmatrix} 0 & 1 & 1 & 0 \end{pmatrix}^{\otimes m}.$$

Thus, $L_m \cdot H_{m\times n}$ is a 1×4^m matrix which is obtained from $H_{m\times n}$ by first deleting the rows whose row index in the $wvux$-order contains at least one letter of u and x, and then adding up all of the remaining non-zero entries by columns. Again, $(L_m \cdot H_{m\times n}) \cdot R_m^t$ is a 1×1 matrix which is obtained from $L_m \cdot H_{m\times n}$ by first deleting the columns whose column index in the $uvwx$-order contains at least one letter of u and x, and then adding up all of the remaining non-zero entries. Thus, $P_{m\times n}(z) = L_m \cdot H_{m\times n} \cdot R_m^t$. By Lemma 2.2,

$$P_{m\times n}(z) = L_m \cdot (V_m + W_m)^n \cdot R_m^t.$$

We finish the proof of Theorem 1.1. ∎

Some values of $P_{m\times n}(1)$ are computed by Matlab and listed in Table 1.

Table 1. $P_{m\times n}(1)$.

	$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$	$m = 6$
$n = 1$	2	4	7	13	24	44
$n = 2$	4	11	33	98	291	865
$n = 3$	7	33	163	803	3971	19587
$n = 4$	13	98	803	6547	53389	435027
$n = 5$	24	291	3971	53389	720417	9706901
$n = 6$	44	865	19587	435027	9706901	216173426
$n = 7$	81	2570	96693	3546870	130854309	4817792042
$n = 8$	149	7637	477297	28911809	1763845523	107354061547
$n = 9$	274	22693	2355925	235681253	23775564134	2392171690343
$n = 10$	504	67432	11629027	1921212987	320481684651	53305366529469
$n = 11$	927	200373	57401721	15661161199	4319920870201	$\approx 1.18781 \times 10^{15}$
$n = 12$	1705	595405	283338413	127665372304	58230152122968	$\approx 2.64682 \times 10^{16}$
$n = 13$	3136	1769236	1398577069	1040691953095	784910642479634	$\approx 5.89796 \times 10^{17}$
$n = 14$	5768	5257255	6903468049	8483425185009	$\approx 1.05802 \times 10^{16}$	$\approx 1.31425 \times 10^{19}$
$n = 15$	10609	15621845	34075967931	69154476414585	$\approx 1.42615 \times 10^{17}$	$\approx 2.92857 \times 10^{20}$
$n = 16$	19513	46420050	168201202963	563727672983607	$\approx 1.92237 \times 10^{18}$	$\approx 6.52579 \times 10^{21}$
$n = 17$	35890	137936399	830252119477	$\approx 4.59535 \times 10^{15}$	$\approx 2.59125 \times 10^{19}$	$\approx 1.45415 \times 10^{23}$
$n = 18$	66012	409875693	4098178655825	$\approx 3.74600 \times 10^{16}$	$\approx 3.49286 \times 10^{20}$	$\approx 3.24031 \times 10^{24}$
$n = 19$	121415	1217938738	20228877377719	$\approx 3.05363 \times 10^{17}$	$\approx 4.70819 \times 10^{21}$	$\approx 7.22045 \times 10^{25}$
$n = 20$	223317	3619084505	99851059281979	$\approx 2.48923 \times 10^{18}$	$\approx 6.34638 \times 10^{22}$	$\approx 1.60894 \times 10^{27}$

3. Conclusions

The concept of dissociation sets emerged in the 1980s as a generalization of independent sets. In the last few decades, it has gained the interest of many scholars. In this paper, we deeply discuss the enumeration problem of dissociation sets in grid graphs and use the state matrix recursion algorithm to calculate the number of dissociation sets in a grid graph.

Using this algorithm, we derive the dissociation polynomial $P_{m\times n}(z)$ for the grid graph $G_{m\times n}$. When $z = 1$, we can precisely calculate the number of dissociation sets in $G_{m\times n}$ for different values of m and n, as presented in Table 1. A notable observation from Table 1 is that as n increases, the size of the matrices obtained and analyzed in Section 2 is significantly smaller than the total number of dissociation sets in $G_{m\times n}$. This demonstrates the superiority of the algorithm.

The state matrix recursion algorithm has been applied to several other enumeration problems. Due to its intuitive nature and wide applicability, the algorithm can be effectively extended to enumeration problems involving other substructures of graph.

Use of AI tools declaration

The authors declare that they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by Beijing Natural Science Foundation (No. 1232005).

Conflict of interest

The authors declare that they have no conflicts of interest.

References

1. F. Bock, J. Pardey, L. D. Penso, D. Rautenbach, A bound on the dissociation number, *J. Graph Theory*, **103** (2023), 661–673. https://doi.org/10.1002/jgt.22940
2. F. Bock, J. Pardey, L. D. Penso, D. Rautenbach, Relating the independence number and the dissociation number, *J. Graph Theory*, **104** (2023), 320–340. https://doi.org/10.1002/jgt.22965
3. N. J. Calkin, H. S. Wilf, The number of independent sets in a grid graph, *SIAM J. Discrete Math.*, **11** (1998), 54–60. https://doi.org/10.1137/S089548019528993X
4. S. Cheng, B. Wu, Number of maximal 2-component independent sets in forests, *AIMS Math.*, **7** (2023), 13537–13562. https://doi.org/10.3934/math.2022748
5. A. Itai, C. H. Papadimitriou, J. L. Szwarcfiter, Hamilton paths in grid graphs, *SIAM. J. Comput.*, **11** (1982), 676–686. https://doi.org/10.1137/0211056
6. F. Kardoš, J. Katrenič, I. Schiermeyer, On computing the minimum 3-path vertex cover and dissociation number of graphs, *Theoret. Comput. Sci.*, **412** (2011), 7009–7017. https://doi.org/10.1016/j.tcs.2011.09.009
7. S. Lomonaco, L. Kauffman, Quantum knots, *Proc. SPIE*, **5436** (2004), 268–284. https://doi.org/10.1117/12.544072
8. S. Lomonaco, L. Kauffman, Quantum knots and mosaics, *Quantum Inform. Process.*, **7** (2008), 85–115. https://doi.org/10.1007/s11128-008-0076-7
9. S. Oh, State matrix recursion method and monomer-dimer problem, *Discrete Math.*, **342** (2019), 1434–1445. https://doi.org/10.1016/j.disc.2019.01.022
10. S. Oh, Number of dominating sets in cylindric square grid graphs, *Graphs Combin.*, **37** (2021), 1357–1372. https://doi.org/10.1007/s00373-021-02323-8

11. S. Oh, S. Lee, Enumerating independent vertex sets in grid graphs, *Linear Alg. Appl.*, **510** (2016), 192–204. https://doi.org/10.1016/j.laa.2016.08.025
12. Y. Orlovich, A. Dolgui, G. Finke, V. Gordon, F. Werner, The complexity of dissociation set problems in graphs, *Discrete Appl. Math.*, **159** (2011), 1352–1366. https://doi.org/10.1016/j.dam.2011.04.023
13. W. Sun, S. Li, On the maximal number of maximum dissociation sets in forests with fixed order and dissociation number, *Taiwanese J. Math.*, **27** (2023), 647–683. https://doi.org/10.11650/tjm/230204
14. G. L. Thompson, Hamiltonian tours and paths in rectangular lattice graphs, *Math. Mag.*, **50** (1977), 147–150. https://doi.org/10.1080/0025570X.1977.11976635
15. J. Tu, Z. Zhang, Y. Shi, The maximum number of maximum dissociation sets in trees, *J. Graph Theory*, **96** (2021), 472–489. https://doi.org/10.1002/jgt.22627
16. M. Xiao, S. Kou, Exact algorithms for the maximum dissociation set and minimum 3-path vertex cover problems, *Theoret. Comput. Sci.*, **657** (2017), 86–97. https://doi.org/10.1016/j.tcs.2016.04.043
17. M. Yannakakis, Node-deletion problems on bipartite graphs, *SIAM J. Comput.*, **10** (1981), 310–327. https://doi.org/10.1137/0210022
18. L. Zhang, J. Tu, C. Xin, Maximum dissociation sets in subcubic trees, *J. Comb. Optim.*, **46** (2023), 8. https://doi.org/10.1007/s10878-023-01076-9